

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: K. MOGI, et al

Serial No.: 10/647,261

Filing Date: August 26, 2003

For: DATA PREFETCHING METHOD

LETTER CLAIMING RIGHT OF PRIORITY

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

February 5, 2004

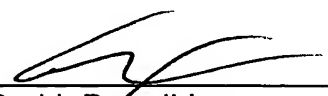
Sir:

Under the provisions of 35 USC 119 and 37 CFR 1.55, applicants hereby claim
the right of priority based on:

**Japanese Application No. 2003-086829
Filed: March 27, 2003**

A Certified copy of said application document is attached hereto.

Respectfully submitted,



Carl I. Brundidge
Registration No. 29,621
ANTONELLI, TERRY, STOUT & KRAUS, LLP

CIB/jdc
Enclosures
703/312-6600

日 本 国 特 許 庁
JAPAN PATENT OFFICE

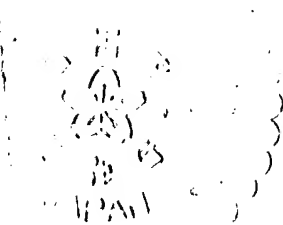
別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 3 月 2 7 日
Date of Application:

出 願 番 号 特 願 2 0 0 3 - 0 8 6 8 2 9
Application Number:
[ST. 10/C] : [J P 2 0 0 3 - 0 8 6 8 2 9]

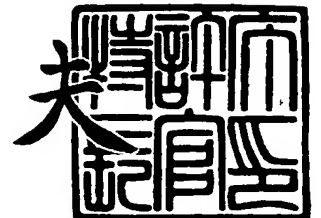
出 願 人 株式会社日立製作所
Applicant(s):



2 0 0 3 年 8 月 2 1 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願

【整理番号】 K03005961A

【あて先】 特許庁長官殿

【国際特許分類】 G06F 12/00

【発明者】

 【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所システム開発研究所内

 【氏名】 茂木 和彦

【発明者】

 【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所システム開発研究所内

 【氏名】 西川 記史

【発明者】

 【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所システム開発研究所内

 【氏名】 出射 英臣

【特許出願人】

 【識別番号】 000005108

 【氏名又は名称】 株式会社 日立製作所

【代理人】

 【識別番号】 100075096

 【弁理士】

 【氏名又は名称】 作田 康夫

【手数料の表示】

 【予納台帳番号】 013088

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

【物件名】 要約書 1
【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データ先読み方法

【特許請求の範囲】

【請求項 1】

データベース管理システムが動作する第一の計算機、前記第一の計算機に接続され、前記データベース管理システムが管理するデータベースのデータを格納し、キャッシュメモリを有する記憶装置及び前記第一の計算機に接続され、前記データベースのデータを使用する第二の計算機とを有する計算機システムにおけるデータの先読みの方法であって、

前記データベース管理システムで実行される処理の内容から所定の条件を満たす処理内容を抽出し、

前記抽出された内容に基づいてデータの先読み方法を決定し、

前記処理の内容が実施される際に、前記記憶装置に前記データの先読み方法に基づくデータの先読みを指示し、

前記処理の内容の実施が終了する際に、前記記憶装置に前記データの先読みの終了を指示することを特徴とするデータ先読み方法。

【請求項 2】

前記所定の条件とは、前記処理の内容に繰り返し実行される部分が含まれているという条件であることを特徴とする請求項 1 記載のデータ先読み方法。

【請求項 3】

前記データの先読みを指示するステップには、前記記憶装置が有するキャッシュメモリで確保される記憶容量を指示するステップが含まれることを特徴とする請求項 2 記載のデータ先読み方法。

【請求項 4】

前記記憶装置に前記データの先読み方法に基づくデータの先読みを指示する際に、前記データベース管理システムに対しても前記データの先読み方法に基づくデータの先読みを指示し、

前記記憶装置に前記データの先読みの終了を指示する際に、前記データベース管理システムに対しても前記データの先読みの終了を指示し、

前記データの先読みを指示するステップには、前記データベース管理システムが有するキャッシュメモリで確保される記憶容量を指示するステップが含まれることを特徴とする請求項 3 記載のデータ先読み方法。

【請求項 5】

前記データの先読み方法を決定するステップでは、データベースの構成に関する情報及び該計算機システムにおける記憶領域のマッピングに関する情報が使用されることを特徴とする請求項 3 記載のデータ先読み方法。

【請求項 6】

前記データ先読み方法を決定するステップでは、前記処理の繰り返し回数の情報を利用してデータ先読み方法を決定することを特徴とする請求項 4 記載のデータ先読み方法。

【請求項 7】

前記第一の計算機と前記第二の計算機が同一の計算機であることを特徴とする請求項 1 記載のデータ先読み方法。

【請求項 8】

データベース管理システムが動作する第一の計算機、前記第一の計算機に接続され、前記データベース管理システムが管理するデータベースのデータを格納し、キャッシュメモリを有する記憶装置及び前記第一の計算機に接続され、前記データベースのデータを使用する第二の計算機とを有する計算機システムにおけるデータの先読みの方法であって、

前記データベース管理システムで処理が実行される際に、前記処理の内容から所定の条件を満たす処理内容を抽出し、

前記抽出された内容に基づいてデータの先読み方法を決定し、

前記記憶装置に前記データの先読み方法に基づくデータの先読みを指示し、

前記処理の内容の実施が終了する際に、前記記憶装置に前記データの先読みの終了を指示することを特徴とするデータ先読み方法。

【請求項 9】

前記第一の計算機と前記第二の計算機が同一の計算機であることを特徴とする請求項 8 記載のデータ先読み方法。

【請求項 1 0】

前記処理内容の抽出ステップ、前記先読み方法の決定ステップ、前記データの先読み指示ステップ及び前記データの先読みの終了を指示するステップが前記第一の計算機で行われることを特徴とする請求項 1 及び 8 に記載されたうちいずれか一つの方法。

【請求項 1 1】

前記処理内容の抽出ステップ、前記先読み方法の決定ステップ、前記データの先読み指示ステップ及び前記データの先読みの終了を指示するステップが前記第二の計算機で行われることを特徴とする請求項 1 及び 8 に記載されたうちいずれか一つの方法。

【請求項 1 2】

前記処理内容の抽出ステップ、前記先読み方法の決定ステップ、前記データの先読み指示ステップ及び前記データの先読みの終了を指示するステップが前記記憶装置で行われることを特徴とする請求項 1 及び 8 に記載されたうちいずれか一つの方法。

【請求項 1 3】

前記処理内容の抽出ステップが前記第二の計算機で行われ、前記先読み方法の決定ステップ、前記データの先読み指示ステップ及び前記データの先読みの終了を指示するステップが前記第一の計算機で行われることを特徴とする請求項 1 及び 8 に記載されたうちいずれか一つの方法。

【請求項 1 4】

データベース管理システムが動作する計算機、

前記データベース管理システムが管理するデータベースのデータを格納し、かつ、キャッシュを有する記憶装置を有する計算機システムで実行されるデータ先読みプログラムであって、

前記データベース管理システムで実行される処理の内容に関する情報を取得するステップと、

前記データベース管理システム、前記計算機、前記記憶装置のそれぞれからデータのマッピングに関する情報を取得するステップと、

処理の開始を示す情報を取得するステップと、
前記取得情報を用いてデータ先読み方法を決定するステップと、
前記記憶装置に対して前記データ先読み方法を与えるステップと、
処理の終了を示す情報を取得するステップと、
記記憶装置に対して前記データ先読み方法の解除を指示するステップとを有することを特徴とするデータ先読みプログラム。

【請求項 1 5】

請求項 1 4 記載のデータ先読みプログラムを記憶する記憶媒体。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、記憶装置へのアクセス性能向上方法、特に、データベース管理システム(DBMS)が稼動する計算機システムの記憶装置におけるデータの先読みによるアクセス性能向上方法に関する。

【0 0 0 2】

【従来の技術】

近年、システムで扱われるデータ量の増大とともに、これらのデータを管理するためのデータベース管理システム(DBMS)は極めて重要なものとなっている。DBMSの性能は、計算機から記憶装置に格納されたデータへのアクセス性能と密接に関係するため、DBMSの性能向上のためには、計算機から記憶装置へのアクセス性能の向上が極めて重要となる。

【0 0 0 3】

一般的に、記憶装置においては、記憶装置内でデータを一時的に保管する高速アクセス可能なデータキャッシュを用意し、データリード時にデータがキャッシュに存在している状態（以下「ヒット」）を作り出すことによりアクセス性能を向上させる手法が用いられる。そのため、使用が予測されるデータを、実際のアクセス要求が来る前に如何にしてキャッシュに予め読み出しておくか（以下「先読み」）が、記憶装置のアクセス性能を向上させる上で重要となる。

【0 0 0 4】

非特許文献1では、オペレーティングシステム（以下「OS」）が、プログラムにより発行されたヒントを用いて計算機上のファイルキャッシュにデータを先読みする機能と、その制御方法について論じられている。非特許文献1では、今後アクセスが行われるファイルとアクセス先領域に関するヒントをプログラムが発行するよう、管理者等によってプログラムが修正される。

【0005】

非特許文献2では、非特許文献1の技術を更に進めた技術が開示されている。ここでは、ヒントを発行するために、I/O処理待ち時に今後実行すると予想される処理を投機的に実行させるようにプログラムに修正を加え、その処理結果を元にヒントが発行される。また、そのプログラムの修正を自動で行うツールに関しても開示されている。

【0006】

非特許文献3では、DBMSがこれから行う問い合わせ処理の実行計画を記憶装置が取得し、それを利用したデータ先読み方法に関する技術が開示されている。処理の実行計画を受け取った記憶装置は、DBMSがある表に対する索引を読んだ後、対応する表のデータを記憶するどのブロックにアクセスされるかが判断できるようになる。そこで、記憶装置は、索引のデータを連続的に読み出し、その索引によりアクセス先が定まる表のデータを保持するブロック群を把握し、それらに対するアクセスをスケジューリングし、効果的に先読みを実施する。特に、記憶装置は、この処理をDBMSが実行される計算機とは独立に実施することができる。

【0007】

【非特許文献1】

R. Hugo Patterson 他著、" Informed Prefetching and Caching、" In Proc. of the 15th ACM Symposium on Operating System Principles、 pp. 79-95、 Dec. 1995.

【非特許文献2】

Fay Chang 他著、" Automatic I/O Hint Generation through Speculative Execution、" the 3rd Symposium on Operating Systems Design and Implementation、 Feb. 1999、

【非特許文献 3】

向井 他著、「高機能ディスクにおけるアクセスプランを用いたプリフェッチ機構に関する評価」、第11回データ工学ワークショップ（DEWS2000）論文集講演番号3B-3、2000年7月発行CD-ROM、主催：電子情報通信学会データ工学研究専門委員会

【0 0 0 8】**【発明が解決しようとする課題】**

DBMS上で実施される処理の中には、同じ形の構造化照会言語（Structured Query Language。以下「SQL」）で記述された処理文（以下「SQL文」）で与えられる処理を多数回、繰り返し実行するものが存在する。この場合、1つの処理に対応する先読みすべきデータを特定することは困難である。しかし、同じ形の処理が多数回実施されることを前提にしたとき、多数回実施される処理でアクセスされる確率が高いデータの記憶領域を判別し、それらを先読みすることができると考えられる。

【0 0 0 9】

しかし、非特許文献 1 では、DBMSによる効果の評価を行っているが、繰り返し同じ形のSQL文による処理を行うことまでは述べられていない。また、非特許文献 2 では、入力データによりアクセスされるデータが変わるときにも効果が出るように、処理の投機的実行結果を利用することが開示されているが、その入力データの特徴（つまり、DBMSにおけるSQL文の特徴）は考慮されていない。

【0 0 1 0】

さらに、非特許文献 3 では、記憶装置に与えられる情報は、実行計画以外は述べられていない。従って、同じ形のSQL文が繰り返されることを識別する情報は送られず、同じ形のSQL文が繰り返し実行されることを前提としたデータの先読みを実施することができない。

【0 0 1 1】

本発明の目的は、DBMSが稼動する計算機システムにおいて、同じ形のSQL文で与えられる処理が多数回、繰り返し実行される際の記憶装置のアクセス性能を向上させることである。

【0012】

【課題を解決するための手段】

本発明では、データの先読みを管理する先読みプログラムが、繰り返し実行されるSQL文に関する情報と、その処理の実行開始情報を取得し、それを基にデータの先読み指示を記憶装置に対して発行する。

【0013】

第一の方法では、繰り返し実行されるSQL文の取得と、その処理内容の解析を先読みプログラムが事前に実施し、先読みすべきデータを事前に把握する。その処理を実行する直前に、処理開始を先読みプログラムに対して通知する。先読みプログラムは、事前解析結果や与えられたキャッシュ量を基に、DBMSや記憶装置に対してキャッシュ量の設定やデータの先読み方法の指示を発行する。先読みプログラムは処理の完了報告を受け、その後に、処理のために割り当てたキャッシュの解放要求をDBMSや記憶装置に発行する。

【0014】

第二の方法では、繰り返し実行されるSQL文を、処理開始時に処理プログラムから先読みプログラムに与える。先読みプログラムは、与えられたSQL文の解析を行い、それと与えられたキャッシュ量の設定を基にDBMSや記憶装置に対してキャッシュ量の設定とデータの先読み方法の指示を発行する。先読みプログラムは繰り返し処理の完了報告を受け、その後に、処理のために割り当てたキャッシュの解放要求をDBMSや記憶装置に発行する。

【0015】

第三の方法では、先読みプログラムはDBMSのフロントエンドプログラムの振舞う。先読みプログラムは通常、処理プログラムからSQL文を受領し、それをDBMSに転送、DBMSから処理結果を受け取り、それを処理プログラムに返す。先読みプログラムに、これから与えられるSQL文が繰り返し処理が実施されることを通知された場合、SQL文の受領後、その解析を行い、それと与えられたキャッシュ量の設定を基にDBMSや記憶装置に対してキャッシュ量の設定とデータの先読み方法の指示を発行、その後にDBMSに対してSQL文を転送するようにする。先読みプログラムが繰り返し処理の完了報告を受た場合は、処理のために割り当てたキャ

ツシュの解放要求をDBMSや記憶装置に発行する。

【0 0 1 6】

同じ形の処理が多数回実施されることを前提にしたとき、アクセスされる確率が高いデータの記憶領域は、その処理で利用されるSQL文の実行計画をDBMSから取得し、それから把握されるデータアクセス先とアクセス方法、アクセス順序から求める。

【0 0 1 7】

【発明の実施の形態】

以下、本発明の実施の形態を説明する。なお、これにより本発明が限定されるものではない。

【0 0 1 8】

まず、第一の実施形態について説明する。第一の実施形態における計算機システムは、繰り返し実行されるSQL文の取得と、処理内容の解析を計算機が先読みプログラムを実施することで事前に行う。その後、繰り返し実行されるSQL文に基づく処理の処理開始通知を契機に、計算機は事前解析の結果を基に先読み指示を記憶装置に発行する。

【0 0 1 9】

図1は、第一の実施形態の計算機システムの構成を示す図である。計算機システムは、記憶装置40、記憶装置40を使用する計算機（以下「サーバ」）70、Jobプログラム100の実行管理を行う計算機（以下「Job管理サーバ」）120、プログラムの開発に使用される計算機（以下「開発サーバ」）140、先読みプログラム160の実行に利用される計算機（以下「先読み制御装置」）170、及び記憶領域の仮想化処理を行う仮想化スイッチ60を有する。各々の装置はネットワークI/F 22を有し、それを介してネットワーク24に接続され、相互に通信可能である。

【0 0 2 0】

サーバ70、仮想化スイッチ60、及び記憶装置40は各々I/OパスI/F 32を有し、それを介して通信線（以下「I/Oパス」）34により接続される。サーバ70と記憶装置40間のI/O処理はI/Oパス34を用いて行われる。尚、I/Oパス34は、装置間で異なる物理媒体や異なるプロトコルでデータ転送を行う通信線が用いられてもよ

い。また、ネットワーク24とI/Oパス34が同一の通信線でもよい。

【0021】

記憶装置40は、CPU12、メモリ14、ディスク装置（以下「HDD」）16、ネットワークI/F22、I/OパスI/F32を有し、それらは内部バス18で接続される。なお、HDD16は単数でも複数でもよい。メモリ14の記憶領域は不揮発領域と高性能領域とに物理的に分割されている。

【0022】

記憶装置40を制御するプログラムである制御プログラム44や先読みプログラム160aは、メモリ14の不揮発領域に記憶され、起動時にメモリ14の高性能領域へ移された後にCPU12により実行される。記憶装置40が有する機能は、後述する先読みプログラム160aによるもの以外は、全て制御プログラム44により制御される。また、記憶装置40は、制御プログラム44を実行することで、ネットワークI/F 22やI/OパスI/F 32を利用して外部装置と通信し、それを利用して先読みプログラム160aも外部と通信可能である。

【0023】

メモリ14には制御プログラム44が記憶装置40を制御・管理するために利用する管理情報46が記憶される。更に、メモリ14の高性能領域の一部は外部装置からアクセス要求のあったデータを一時的に記憶しておく領域であるデータキャッシュ42に割り当てられる。このとき、HDD16に未書き込みなデータ等、高信頼性を必要とするデータをメモリ14の不揮発領域に記憶してもよい。

【0024】

記憶装置40は、HDD16が有する物理記憶領域を仮想化して1又は複数の論理ディスク装置（以下、「LU」と称す）208を外部装置に対して提供する。LU208は、HDD16と一対一に対応してもよいし、複数のHDD16から構成される記憶領域と対応してもよい。また、1つのHDD16が複数のLU208に対応してもよい。その対応関係は、管理情報46中に領域マッピング情報310の形で保持される。

【0025】

記憶装置40は、キャッシュ指示730中のデータ領域、キャッシュ量の情報に基づいて、LU208の指定された領域に関して、指定された量のデータキャッシュ42

内の記憶領域の割り当て設定・解除を行う。このキャッシュの設定・解除は動的（以下「他の処理を停止させることなく実施される」の意味で利用）に行える。記憶装置40は、キャッシュ指示730に含まれるグループ化に関する値が同じものに関しては、1つの領域として管理する。

【 0 0 2 6 】

また、計算機システムの使用者等は、キャッシュ指示730に含まれるキャッシュ方法の指示により、それらのデータキャッシュ42の領域に対してデータを即時に先読みすること（以下「即時先読み」）、あるいは、その領域に関しては、全てのアクセス要求が連続的に続くとして先読みすること（以下「シーケンシャル」）、あるいは、現在の設定設定の解除（以下「設定解除」）を外部装置から記憶装置40に指示することができる。また、記憶装置40は、キャッシュ指示730中のアクセス順の情報に基づいて、先読みする順番を判断する。なお、キャッシュ指示730は、先読みプログラム160aから与えられる。

【 0 0 2 7 】

仮想化スイッチ60は、CPU12、メモリ14、ネットワークI/F22、I/OパスI/F32を有し、それらは内部バス18で接続される。メモリ14の記憶領域は不揮発領域と高性能領域とに物理的に分割されている。

【 0 0 2 8 】

仮想化スイッチ60を制御するプログラムである制御プログラム64や先読みプログラム160bは、メモリ14の不揮発領域に記憶され、起動時にメモリ14の高性能領域へ移された後にCPU12により実行される。仮想化スイッチ60が提供する機能は、制御プログラム64により制御される。また、仮想化スイッチ60は、制御プログラム64を実行して、ネットワークI/F 22やI/OパスI/F 32を利用して外部装置と通信し、それを利用して先読みプログラム160bも外部と通信可能である。

【 0 0 2 9 】

また、メモリ14には制御プログラム64が仮想化スイッチ60を制御・管理するために利用する管理情報66が記憶される。

【 0 0 3 0 】

仮想化スイッチ60は、本装置に接続された記憶装置40から提供されるLU 208を

認識し、その記憶領域を仮想化して仮想ボリューム206を外部装置に提供する。尚、仮想化スイッチ60が多段接続された場合には、仮想化スイッチ60は、他の仮想化スイッチ60が提供する仮想ボリューム206を記憶装置40から提供されるLU 208と等価に扱い、その記憶領域を仮想化して仮想ボリューム206を外部装置に提供する。LU208と仮想ボリューム206との対応関係は、管理情報66中に領域マッピング情報310の形で保持される。

【 0 0 3 1 】

サーバ70は、CPU12、メモリ14、HDD16、ネットワークI/F 22及びI/OバスI/F 32を有し、それらは内部バス18で接続される。メモリ14には、OS72及び先読みプログラム160cがHDD16から読み込まれ、CPU12により実行される。先読みプログラム160cの詳細に関しては後述する。

【 0 0 3 2 】

OS72は、サーバ70上で実行されるプログラム対して、ネットワークI/F 22、I/OバスI/F 32等のハードウェア制御や、ネットワーク24を介した他の装置との通信、I/Oバス34を通してのデータ転送処理、複数プログラム間の実行制御、外部装置で実行されるプログラムを含む複数プログラム間でのメッセージ交換、外部装置からのプログラムの起動要求受付等、基本的な処理を提供するためにCPU12で実行されるプログラム群であり、ボリュームマネージャ78、ファイルシステム80を含む。メモリ14に読み込まれたOS72には、そのOS72や他のOS72を構成するプログラムが利用する管理情報であるOS管理情報74が含まれる。OS管理情報74は、サーバ70のハードウェア構成の情報を含む。OS72は、OS管理情報74中に記憶されている情報を外部プログラムが読むためのソフトウェアインターフェイスを有する。なお、本図では、サーバ70は1つのファイルシステム80しか有していないが、複数のファイルシステム80を有してもよい。

【 0 0 3 3 】

ボリュームマネージャ78は、記憶装置40から提供されるLU208や仮想化スイッチ60から提供される仮想ボリューム206の記憶領域をさらに仮想化した論理ボリューム204をファイルシステム80に提供するためにサーバ70で実行されるプログラムである。仮想ボリューム206と論理ボリューム204との対応関係は、OS管理情

報74中に領域マッピング情報310の形で保持される。

【 0 0 3 4 】

ファイルシステム80は、記憶装置40から提供されるLU208や仮想化スイッチ60から提供される仮想ボリューム206、ボリュームマネージャ78から提供される論理ボリューム204の記憶領域を仮想化し、ファイル202を他のプログラムに提供するためにサーバ70で実行されるプログラムである。ファイル202と論理ボリューム204等との対応関係は、OS管理情報74中に領域マッピング情報310の形で保持される。なお、ファイル202と同じソフトウェアインターフェイスで、論理ボリューム204、仮想ボリューム206、LU 208の記憶領域に直接アクセスするローデバース機能もファイルシステム80により提供されたとする。

【 0 0 3 5 】

DBMS90は、DBに関する一連の処理・管理を実行するためにサーバ70で実行されるプログラムである。本プログラムは、HDD16もしくは記憶装置40からメモリ14に読み出されてCPU12により実行される。メモリ14上に読み込まれたDBMS90は、DBMS90の管理情報であるDBMS管理情報92を有し、その中に、DBMS90が利用・管理する表・索引・ログ等（以下、まとめて「データ構造」と称する）の記憶領域の管理情報であるデータ記憶領域情報510を含む。また、サーバ70は、DBMS90を実行することで、メモリ14の領域をキャッシュ94として利用し、その最低利用量をデータ構造毎に管理する。DBMS90は、DBMS管理情報92を外部プログラムが読むためのソフトウェアインターフェイスを有する。また、DBMS90は、与えられたSQL文700による処理の実行計画570を出力するソフトウェアインターフェイスを有する。

【 0 0 3 6 】

尚、一般的に、一つの計算機においては複数のプログラムが並行して実行され、これらのプログラム間でメッセージの遣り取りを行うことで協調して処理が行われる。したがって、実際には一つのCPU（あるいは複数）で複数のプログラムが実行され、メッセージの遣り取りはOS72により管理されたメモリ14上の領域等を介して行われる。しかし、説明の簡略の為、本明細書においては、上記のようなメッセージの遣り取り等を、CPUで実行されるプログラムを主語（又は目的語

）にして説明する。

【 0 0 3 7 】

Jobプログラム100は、ユーザが行う業務のためにサーバ70上で実行されるプログラムである。Jobプログラム100は、DBMS90に対して処理要求を発行する。Jobプログラム100は、Job管理プログラム130がネットワークを通してOS72に起動要求を発行し、HDD16もしくは記憶装置40からメモリ14に読み出されてCPU12により実行される。

【 0 0 3 8 】

なお、Jobプログラム100は記憶装置40に記憶されるデータを扱う際に常にDBMS90に処理要求を発行してもよく、この場合には、Jobプログラム100が実行されるサーバ70はI/OパスI/F32を有さなくてもよい。なお、Jobプログラム100には、ソースコードから実行形式に変換したものをを用いてもよいし、SQL文をベースとした処理言語（以下「SQLスクリプト」）を用いて記述されたものを実行時にスクリプト実行プログラムに与えて、スクリプト実行プログラムがそれを解釈しながら実行する形式を採用しても良い。

【 0 0 3 9 】

DBMS90やJobプログラム100は、1台のサーバ70上で複数同時に実行することができる。また、DBMS90とJobプログラム100が異なるサーバ70上で実行されてもよく、その場合には、Jobプログラム100はDBMS90に処理要求をネットワーク24を経由して伝達する。

【 0 0 4 0 】

Job管理サーバ120は、CPU12、メモリ14、HDD16、CD-ROMドライブ20及びネットワークI/F 22を有し、それらは内部バス18で接続される。メモリ14には、OS72、Job管理プログラム130及び先読みプログラム160dがHDD16から読み込まれ、CPU12により実行される。先読みプログラム160 dの詳細に関しては後述する。

【 0 0 4 1 】

Job管理プログラム130は、Job管理サーバ120が有するJob管理機能を実現するプログラムであり、その機能を実現するために必要な管理情報であるJob管理情報132をメモリ14に有する。

【 0 0 4 2 】

開発サーバ140は、CPU12、メモリ14、HDD16、及びネットワークI/F 22を有し、それらは内部バス18で接続される。メモリ14には、OS72、開発管理プログラム150、及び先読みプログラム160eがHDD16から読み込まれ、CPU12により実行される。先読みプログラム160eの詳細に関しては後述する。

【 0 0 4 3 】

開発プログラム150は、Jobプログラム100を開発するためのシステムの管理者等が使用するプログラムである。開発プログラム150は、Jobプログラム100のソースコードやその他のプログラム開発に必要な情報を含む開発コード152を開発サーバ140内のHDD16中に記憶する。

【 0 0 4 4 】

先読み制御装置170は、CPU12、メモリ14、HDD16及びネットワークI/F 22を有し、それらは内部バス18で接続される。メモリ14には、OS72及び先読みプログラム160fがHDD16から読み込まれ、CPU12により実行される。先読みプログラム160fの詳細に関しては後述する。なお、先読み制御装置170は必ずしも存在しなくともよい。

【 0 0 4 5 】

キーボード・マウス等の入力装置112および表示画面114を有する管理端末110がネットワーク24を介して接続される。この接続は、ネットワーク24とは異なる通信線を用いてもよい。管理者は、原則、管理端末110を介して各種計算機に各種指示を発行したり、その他の処理を行う。

【 0 0 4 6 】

OS72、DBMS90、Jobプログラム100、開発プログラム150及び先読みプログラム160c、160d、160e、160fは、それらを記憶したCD-ROM（記憶媒体）から管理サーバ120が有するCD-ROMドライブ20を用いて読み出され、ネットワーク24を介してサーバ70、管理サーバ120、開発サーバ150、先読み制御装置170内のHDD16もしくは記憶装置40にインストールされる。

【 0 0 4 7 】

なお、本図においては、Job管理プログラム130、開発プログラム150がサーバ7

0とは異なる計算機で実行されるとしているが、これらのプログラムがサーバ70上で実行されてもよい。Job管理プログラム130がサーバ70で実行される場合、CD-ROMドライブ20は、いずれかのサーバ70が保持し、各種プログラムのインストールに利用されるものとする。

【 0 0 4 8 】

図2は、第一の実施形態におけるDBMS90が管理するデータのデータマッピングの階層構成を示す図である。本図では、サーバ70と記憶装置40との間に1つの仮想化スイッチ60が存在する場合を説明する。以下、ある2つの階層について、DBMS90に近い方を上位、HDD16に近い方を下位の階層と称する。ファイル202、論理ボリューム204、仮想ボリューム206、LU 208をまとめて「仮想構造」と称し、更に、仮想構造にHDD16を加えたものをまとめて「管理構造」と称する。また、仮想構造を提供する、記憶装置40、仮想化スイッチ60、ボリュームマネージャ78及びファイルシステム80をまとめて「仮想化機構」と称する。

【 0 0 4 9 】

図2では、DBMS90は、それが管理しているデータ構造200を記憶しているファイル202に対してアクセスを行う。ファイル202はファイルシステム80により提供され、ファイルシステム80は、ファイル202に対するアクセスを対応する論理ボリューム204の領域へのアクセスに変換する。ボリュームマネージャ78は、論理ボリューム204に対するアクセスを対応する仮想ボリューム206の領域へのアクセスに変換する。仮想化スイッチ60は、仮想ボリューム206に対するアクセスを対応するLU 208の領域へのアクセスに変換する。記憶装置40は、LU208に対するアクセスを、対応するHDD16に対するアクセスに変換する。このように、仮想化機構は、それが上位階層に提供する仮想構造のデータを下位階層に存在する1つ以上の管理構造の記憶領域にマッピングする。

【 0 0 5 0 】

ある仮想構造のデータがHDD16にマッピングされる経路が複数存在してもよい。あるいは、ある仮想構造のデータの同一部分が複数の下位階層の管理構造にマッピングされてもよい。これらの場合には、仮想化機構がそのようなマッピングであることが領域マッピング情報310中に保持される。

【 0 0 5 1 】

また、ある管理構造が複数のサーバ70に共有されるマッピングを有してもよい。これは、フェイルオーバ構成をとるサーバ70とそのサーバ70で実行されるDBMS 90において利用される。

【 0 0 5 2 】

本実施形態では、論理層212における管理構造間のデータの対応関係が明確化されればよく、サーバ70でボリュームマネージャ78が使用されなくてもよい。仮想化スイッチ60は複数段存在してもよいし、仮想化スイッチ60が存在せずにサーバ70と記憶装置40がI/Oパス34により直結されてもよい。仮想化スイッチ60に相当するスイッチが記憶領域の仮想化機能を有しない場合、サーバ70と記憶装置40が直結されているのと等価である。仮想化スイッチ60が存在しない、もしくは、仮想化スイッチ60に相当するスイッチが記憶領域の仮想化機能を有しない場合、先読みプログラム160bは存在しなくてもよい。

【 0 0 5 3 】

以下、各装置やプログラムが保持するデータ構造に関して説明する。

【 0 0 5 4 】

図 3 は、領域マッピング情報310のデータ構造を示す図である。領域マッピング情報310は、仮想化機構が提供する仮想構造の領域と、それが利用する管理構造の領域の対応関係を保持するものであり、エントリ312及び314を有する。エントリ312には、仮想化機構が上位階層に提供する仮想構造の領域に関する情報が登録される。具体的には、エントリ312は、仮想構造の識別子である仮想構造IDを保持するエントリ及びその構造内の領域を示すエントリの組を有する。エントリ314には、エントリ312に対応する下位階層の管理構造の領域に関する情報が登録される。具体的には、エントリ314は、管理構造を提供する仮想化機構の識別子である仮想化機構IDを保持するエントリ、管理構造の識別子である管理構造IDを保持するエントリ及びその構造内領域を示すエントリの組を有する。なお、記憶装置40においては、仮想化機構IDを有するエントリを保持しない。

【 0 0 5 5 】

前述のように、異なる仮想構造が同一の管理構造の記憶領域を利用することが

許される。また、仮想化機構ID、仮想構造ID及び管理構造IDはシステム内で一意に定まる識別子であるとする。そうでない場合でも、装置の識別子を付加することによりシステム内で一意に定まるようにすることができる。

【0056】

図4は、DBMS管理情報92中に保持されるデータ記憶領域情報510のデータ構造を示す図である。データ記憶領域情報510は、DBMS90が管理するデータの記憶領域管理に用いられる。データ記憶領域情報510は、データ構造の名前であるデータ構造名を保持するエントリ512及び対応するデータ構造がファイル202のどの位置に記憶されているかの情報であるデータ記憶位置を保持するエントリ514の組からなる。なお、データ構造名は、DBMS90内で一意に定まる名前であるとし、DBMS90内でDB毎に同じ名前が許される場合には、DBの識別子も含めたものをデータ構造名として利用する。

【0057】

図5は、DBMS管理情報92中に保持される表データ量情報520のデータ構造を示す図である。表データ量情報520は、表のデータ量管理に用いられる情報である。表データ量情報520は、表のデータ構造名を保持するエントリ521、その表におけるデータページの大きさに関する情報であるデータページサイズを保持するエントリ522、その表が利用しているデータページ数を保持するエントリ524及びそのデータが利用可能なキャッシュ94の最低量に関する情報であるキャッシュ量を保持するエントリ526を有する。

【0058】

図6は、DBMS管理情報92中に保持される索引情報530のデータ構造を示す図である。検索情報530は、DBMS90の索引の管理に用いられる情報である。索引情報530は、索引のデータ構造名を保持するエントリ531、その索引が付加された表のデータ構造名である対応表名を保持するエントリ532、索引種別を保持するエントリ534、データページサイズを保持するエントリ533、データページ数を保持するエントリ535、データページのうち、B-Tree索引の場合にリーフノードのデータを保持しているデータページ数であるLeafノードページ数を保持するエントリ536、その索引の最低限利用可能なキャッシュ量を保持するエントリ537、その索

引を利用して検索が行われる属性の属性名の組である検索属性を保持するエントリ538及び検索属性における1回の検索で得られると期待されるタプル数の情報である期待タプル数を保持するエントリ542の組からなる。なお、1つの索引に複数の検索属性とそれに対応する期待タプル数が存在することがある。また、期待タプル数は、対応する表のデータ解析により得られる値で、平均値や最頻値、あるいは各種指標から計算した値が用いられる。

【0 0 5 9】

図7は、Job管理情報132中に保持されるJob実行管理情報360のデータ構造を示す図である。Job実行管理情報360は、Job管理プログラム130がJobプログラム100の実行を管理する際に利用される。Job実行管理情報360は、実行されるJob毎に保持される。

【0 0 6 0】

Job実行管理情報360は、Jobの識別子であるJob IDを保持するエントリ362、Jobとして実行されるJobプログラム100の識別子であるプログラムIDを保持するエントリ338、Jobの実行開始の条件である実行条件を保持するエントリ364、Jobを実行するサーバ70の識別子であるサーバIDを保持するエントリ332及びそのサーバ70で実行されるコマンドを保持するエントリ368の組、Job依存入力情報を保持するエントリ370、Job依存出力データ情報を保持するエントリ380並びにキャッシュ量情報を保持するエントリ340を含む。

【0 0 6 1】

Job依存入力情報は、そのJobを実行する際に利用されるデータに関する情報である。エントリ370はさらに、利用されるデータを出力する前段のJobのJob IDを保持するエントリ372及びその入力データの識別子であるデータIDを保持するエントリ374の組を有する。

【0 0 6 2】

Job依存出力データ情報は、他のJob実行に利用される本Jobの出力データに関する情報である。エントリ380はさらに、出力データを利用するJobのJob IDを保持するエントリ382及びその出力データの識別子であるデータIDを保持するエントリ374の組を有する。

【 0 0 6 3 】

キャッシュ量情報は、Job開始時にJobプログラム100の実行に際し、DBMS90や記憶装置40において本処理でアクセスされるデータ向けに最低限利用可能なキャッシュ量に関する情報である。エントリ340は更に、処理が実施されるDBMS90の識別子であるDBMS IDを保持するエントリ334及びそのDBMS90で利用可能なキャッシュ94の量に関する情報であるキャッシュ量を保持するエントリ342の組並びに処理に利用されるデータを保持する記憶装置40の識別子である装置IDを保持するエントリ336及びそこで利用可能なデータキャッシュ42の量であるキャッシュ量を保持するエントリ342の組を含む。なお、キャッシュ量情報340は必ずしも保持しなくともよい。

【 0 0 6 4 】

以下、本実施の形態で利用する先読みプログラム160について説明する。先読みプログラム160は、各装置で実行される先読みプログラム160a、160b、160c、160d、160e、160fを構成要素として実現されるものである。複数装置間に存在する先読みプログラム160の構成要素の間では、ネットワーク24を通して必要な情報が交換される。以下で説明する各機能モジュールの処理に関しては、原則、どの装置で実現されてもよく、それら自体が複数の装置に分割して実現されてもよい。

【 0 0 6 5 】

ただし、他のプログラムからの情報・処理状態の取得や、処理の指示・依頼を行う部分に関しては、記憶装置40の制御プログラム44に対しては先読みプログラム160aが、仮想化スイッチ60の制御プログラム64に対しては先読みプログラム160bが、サーバ70のOS72、ボリュームマネージャ78、ファイルシステム80及びDBMS90に対しては先読みプログラム160cが、Job管理サーバ120のJob管理プログラム130に対しては先読みプログラム160dが、開発サーバ140の開発プログラム150に対しては先読みプログラム160eが行う。

【 0 0 6 6 】

ただし、これらの機能は、OS72等が提供するより汎用的なプログラム機能により代行させることが可能であり、その場合には、対応する先読みプログラム160a

、160b、160c、160d、160eが実行されなくともよい。また、先読みプログラム160a、160b、160c、160d、160e、160fは、他のプログラムの機能、特に、DBMS90やJob管理プログラム130の一部として実現されてもよい。

【0067】

図8は、本実施形態における、先読み処理に関する先読みプログラム160やその他プログラム、及びこれらのプログラム間で交換される情報の流れを示した図である。先読みプログラム160は、機能モジュールとして、SQL解析モジュール252、先読み方法決定モジュール254、先読み指示モジュール256、及び情報取得モジュール258を含む。尚、機能モジュールとは、一つのプログラムにおいて、ある処理に特化したサブプログラムやルーチン等を指す。

【0068】

また、先読みプログラム160は、処理情報として、システム情報300及びSQL解析情報280を有し、それらは任意の先読みプログラム160a、160b、160c、160d、160e、160fが実行されている装置のメモリ14上に保持される。先読み方法720は、先読みプログラム160内の機能モジュール間で交換される情報である。以下、利用する情報やその利用法等の詳細を説明していく。尚、以下の説明で、図8に記載した番号を使用する。

【0069】

図9は、先読みプログラム160が事前に実施する情報収集処理の手順を示す図である。なお、本処理の実施前に、先読みプログラム160に先読み指示を発行するJobプログラム100が利用するDBに関しては、その定義が完了し、実際にデータが存在しているものとする。(ステップ2101)

まず、先読みプログラム160の情報取得モジュール258は、先読み指示を発行するJobプログラム100及びそのJobプログラム100が利用するDBに関する情報である先読みJob情報350を管理端末110を介して管理者から受領し、それをシステム情報300中に保存する。

【0070】

図10は、先読みJob情報350のデータ構造を示す図である。先読みJob情報350は、先読み指示を行うJobプログラム100に関する情報として、そのプログラムの

プログラムIDを保持するエントリ421を含む。また、Jobプログラム100が利用するDBの情報として、そのDB管理するDBMS90が実行されるサーバ70のサーバID保持するエントリ422、そのDBMS90のDBMS IDを保持するエントリ423、表データ順情報を登録するエントリ420及び入力相関情報を登録するエントリ430を含む。なお、エントリ420や430は先読みJob情報350に含まれなくともよい。

【 0 0 7 1 】

また、本図は、Jobプログラム100が1つのDBMS90により管理されるDBのデータのみを利用する場合を示している。Jobプログラム100が複数のDBMS90により管理されるDBのデータを利用する場合には、先読みJob情報350にはエントリ422及び423が組にして保持され、更に、エントリ420及び430では、データ構造名に対応するDBMS IDを保持するエントリが付加される。

【 0 0 7 2 】

表データ順情報は、Jobプログラム100が利用するデータのDBMS90からみたデータ順に関する情報である。エントリ420には、利用されるデータ（表）のデータ構造名を保持するエントリ425及びそのデータの並び方に関する情報であるデータ順を保持するエントリ424の組が含まれる。ここで、エントリ424には、“表のある属性によりソートされている”あるいは“挿入(Insert)処理順に記憶される”等の情報が登録される。

【 0 0 7 3 】

入力相関情報は、Jobプログラム100への入力データが、特定のデータ構造のデータ順と同じ順序にソートされていることを示す情報である。エントリ430には、入力データのデータIDを登録するエントリ431及びその入力データと同じ並び順のデータ構造名を登録するエントリ432の組が含まれている(ステップ2102)。

【 0 0 7 4 】

続いて、情報取得モジュール258は、アクセスされるデータの情報や、そのデータのマッピングに関する情報を収集する。まず、ステップ2102で取得した先読みJob情報350中に示されるDBMS IDにより識別されるDBMS90から、データ記憶領域情報510、表データ量情報520、索引情報530からなるDBMS構成情報500を取得し、DBMS IDと共にシステム情報300中に保存する。

【 0 0 7 5 】

続いて、情報取得モジュール258は、DBMS-IDに対応するDBMS90が実行されているサーバ70のファイルシステム80及びボリュームマネージャ78がOS管理情報72中に保持している領域マッピング情報310を取得し、その管理元がわかる識別子とともにシステム情報300中に保存する。また情報取得モジュール258は、順次取得した領域マッピング情報310を判別して、対応する記憶領域を提供する仮想化スイッチ60や記憶装置40から領域マッピング情報310を取得してその管理元がわかる識別子とともにシステム情報300中に保存する(ステップ2103)。

【 0 0 7 6 】

続いて、SQL解析モジュール252は、先読みJob情報350で指定されるJobプログラム100が発行するSQL文に関する情報である抽出SQL情報820を開発プログラム150から取得する。抽出SQL情報820は開発プログラム150中のSQL文抽出モジュール270が、開発コード152を元に作成するものであり、対応するJobプログラム100のプログラムIDとSQL情報から構成される。

【 0 0 7 7 】

なお、プログラムIDを指定して抽出SQL情報820の作成を開発プログラム150に依頼し、それを先読みプログラム160に与える処理は、管理者が行っても良いし先読みプログラム160中のSQL解析モジュール252が直接行っても良い。

【 0 0 7 8 】

SQL文抽出モジュール270は、与えられたプログラムIDで識別されるプログラムに対応する開発コード152に含まれるプログラムのソースコードを元に以下のような処理を行う。

【 0 0 7 9 】

図11は、本実施形態におけるSQL文抽出モジュール270の処理例として、C言語で書かれたソースコードに埋め込みSQL文が含まれている場合の抽出処理の例を示す図である。ソースコードの範囲5002で示される部分においては、for文により繰り返し処理が行われ、その中で幾つかのSQL文が実行される。SQL文抽出モジュール270はこの繰り返し構造を識別し、その中にSQLが存在していることから、そのSQL文が繰り返し実行されると判断し、それに対応するSQL情報として、情

報5000を作成する。情報5000には、繰り返し開始を示す情報5012、範囲5002中の繰り返し実行される埋め込みSQL文を抜き出した情報5010及び繰り返し終了を示す情報5018が含まれる。

【 0 0 8 0 】

なお、情報5012中には、“ LABEL ” という指示子の後に、それぞれの繰り返し処理を識別するための情報5014が付加される。ソースコードの他の部分に関しても同様に繰り返し構文とそこに存在するSQL文を判別し、情報5000と同様なSQL情報を作成する。

【 0 0 8 1 】

図 1 2 は、本実施形態におけるSQL文抽出モジュール270の処理例として、ソースコードがSQLスクリプトで記述された場合の抽出処理の例を示す図である。この例では、範囲5102でカーソルの定義が行われ、範囲5104でカーソルを用いて読み出されたデータ毎に、範囲5106の処理が繰り返し実行される。SQL文抽出モジュール270は範囲5104の繰り返し構造を判別し、対応するSQL情報として情報5100を作成する。情報5100には、繰り返し開始を示す情報5012、範囲5104中の実際に繰り返し実行されるSQL文を抜き出した情報5110及び繰り返し終了を示す情報5018が含まれる。

【 0 0 8 2 】

その他にも独自の処理フローを示すデータ構造を示すものを用いた場合でも、SQL文抽出モジュール270は処理の繰り返し構造を把握し、同様なSQL情報を作成する。

【 0 0 8 3 】

尚、SQL文抽出モジュール270は、繰り返し構造が入れ子状態になっている場合には、最も外側のもののみを繰り返し構造として把握する。また、複数の独立した繰り返し構造が存在する場合には、実行される順にその繰り返し構造に対応するSQL情報を作成する。また、繰り返し構造外にあるSQL文に関しても、情報5012、5018と同様な形式でその旨を示す領域を明示し、繰り返し構造内にある場合と同様にSQL情報を作成してもよい。

【 0 0 8 4 】

なお、繰り返し処理を識別するための情報5014は、プログラム実行時に繰り返し回数を判別するための識別子として利用することができる。そのため、必要に応じて、開発プログラム150や管理者が、抽出SQL文820に含まれる情報5014を、それにより識別される繰り返し処理を駆動するデータのデータIDに更新しても良い。(ステップ2104)

続いて、SQL解析モジュール252が、取得した抽出SQL情報820から、ステップ2501から始まる処理を実行してSQL解析詳細情報290を作成し、SQL解析情報280中に記憶する(ステップ2105)。その後、処理を完了する(ステップ2106)。

【 0 0 8 5 】

図 1 3 は、SQL解析モジュール252が、抽出SQL情報820からSQL解析詳細情報290を作成する処理の手順を示す図である。まず、処理開始時、先読みJob情報350に対応する抽出SQL情報820がSQL解析モジュール252に与えられる(ステップ2501)。

【 0 0 8 6 】

抽出SQL情報820が与えられたSQL解析モジュールは、SQL解析詳細情報290を初期化する。図 1 4 は、SQL解析詳細情報290のデータ構造を示す図である。SQL解析詳細情報290には、対応するJobプログラム100の識別子であるプログラムIDを保持するエントリ281、処理が利用するDBを管理するDBMS90のDBMS IDを保持するエントリ291、繰り返し実行されるSQL文のグループの識別子である繰り返しグループIDを保持するエントリ282、そのグループ間での処理の実行順序を示す実行順を保持するエントリ284、その繰り返し処理を駆動するデータのデータIDである駆動データIDを保持するエントリ286、アクセスを行うデータのデータ構造名を保持するエントリ287、そのデータへのアクセス方法を示すアクセス方法を保持するエントリ288、アクセス方法にランダムアクセスを実行する方法が指定された場合に、1回の処理でアクセスされると期待されるデータページ数を示す期待アクセスページ数を保持するエントリ292、及びシーケンシャルにアクセスされることが期待される場合にその値が“Y”に設定されるシーケンシャルヒントを保持するエントリ294の組が含まれる。

【 0 0 8 7 】

なお、本図では、Jobプログラム100が1つのDBMS90により管理されるDBのデー

タのみを利用する場合を示している。複数のDBMS90により管理されるDBのデータを利用する場合には、SQL解析詳細情報290には、DBMS IDを保持するエントリが全体で1つ保持されるのではなく、DBMS IDとデータ構造名との組が保持されるようにする。

【0088】

SQL解析モジュール252は、プログラムIDをエントリ281に設定し、その他のデータを保持するエントリをクリアすることでSQL解析詳細情報290を初期化する(ステップ2502)。

【0089】

次に、SQL解析モジュール252は、抽出SQL情報820中のSQL情報から、繰り返しグループを把握する。繰り返しグループは、繰り返し開始を示す情報5012とそれに対応する繰り返し終了を示す情報5018により囲まれた部分として把握される。

【0090】

尚、繰り返しグループは複数存在する可能性があるが、その場合、複数のグループは、処理が実施される順に並んでいる。そこで、SQL解析モジュール252は、それぞれのグループに独立した識別子である繰り返しグループIDを付加し、その出現順に実行順を設定し、各々をエントリ282、284に登録する。また、情報5014により指示されるラベルを駆動データIDとしてエントリ286に設定する。なお、抽出SQL情報820中のSQL情報に、繰り返し構造外であることが情報5012、5018と同様な形式で示され含まれている場合には、それらも繰り返しグループとして設定されてもよい(ステップ2503)。

【0091】

その後、SQL解析モジュール252は、それぞれの繰り返しグループ中で、繰り返し開始／終了を示す情報5014／情報5018に挟まれた部分に存在する、繰り返しグループ中で実行されるSQL文700を、本処理に対応するDBMS90に与え、DBMS90から実行計画570を取得する。

【0092】

図15は、本実施形態においてSQL解析モジュール252が取得する実行計画570のデータ構造を示す図である。実行計画570の内容は、幾つかの詳細な処理手順

に分割され、分割された処理手順を個々のノードとした木構造で表現される。この木構造では、個々の処理手順で行われる処理に利用されるデータの依存関係が枝となり、先に実行される処理ほど末端に位置する。また、ノードで複数のデータが利用される場合、そのノードは複数の枝を保持する。

【 0 0 9 3 】

実行計画570には、個々の処理手順を表すノードのノード名を保持するエントリ572、そのノードの親ノードのノード名を保持するエントリ574、そのノードで行われる処理内容を保持するエントリ576、そのノードでデータへのアクセスを行う場合に、そのアクセス先となるデータのデータ構造名を保持するエントリ578及びノードで実施される選択処理の条件等を保持するエントリ582の組が保持される。

【 0 0 9 4 】

ノードで行われる処理としては、表データの全走査、索引へのアクセス、索引参照結果を利用した表へのアクセス、データの選択処理、結合・ソート・総和等の演算等があり、これらを指定する情報がエントリ576に保持される。例えばハッシュ結合演算を行うノードの場合、ビルドフェーズに利用されるデータとプローブフェーズで利用されるデータに対応する枝が存在することになる。ここで、大小関係が存在するようにノード名が付加され、その大小関係を利用してその情報が保持される。

【 0 0 9 5 】

SQL解析モジュール252は、取得した実行計画570中のエントリ576及び578に登録されたノード処理内容及びアクセスデータ構造名から、繰り返しグループ中のSQL文700によりアクセスされるデータ構造とそのアクセス方法を把握し、SQL解析詳細情報290の対応するエントリ287及び288にデータ構造名及びアクセス方法の情報を設定する。SQL解析モジュール252は、これらの処理をステップ2503で把握した全ての繰り返しグループに関して実施する(ステップ2504)。

【 0 0 9 6 】

更に、SQL解析モジュール252は、ステップ2504で把握したアクセスされるデータ構造で、B-Tree索引、もしくは、B-Tree索引を利用してアクセスされる表デー

タに関して、SQL解析詳細情報290中の対応するエントリ292に期待アクセスページ数を設定する。

【0 0 9 7】

具体的にはまず、SQL解析モジュール252は、実行計画570から、処理手順を表す木構造のリーフに位置するノードで、B-Tree索引をアクセスする処理を行うものを把握してそのノードのエントリ582を参照し、そのノードの検索条件を求める。まず、SQL解析モジュール252は、その選択すべき値が他の処理の結果ではなく、SQL文700により一意に指定されているものに関し、システム情報300中に保存されている索引情報530のエントリ542を参照してその検索条件における期待タプル数を求める。その値が、その索引を用いてアクセスされるデータの期待タプル数となる。また、この索引アクセスの基となるデータの期待タプル数は1と定義する。

【0 0 9 8】

その後、SQL解析モジュール252は、再度、B-Tree索引をアクセスする処理を行うノードにおける検索条件を調べ、これまでアクセスされる期待タプル数が求まったデータを用いて検索処理を行うものに関して、駆動データ1タプルあたりに検索される期待タプル数を索引情報530のエントリ542から求める。索引参照を駆動するデータの期待タプル数と、索引参照結果による期待タプル数の積が、その索引を利用してアクセスされるデータのアクセスされる期待タプル数となる。以下繰り返しこの確認を行う。

【0 0 9 9】

B-Tree索引を用いた検索処理によりアクセスされるデータの期待タプル数を前述の方法により可能な範囲で求めた後、SQL解析モジュール252は、基本的に各タプルは異なるデータページに存在すると考え、アクセスされるデータページ数を求める。ただし、あるB-Tree索引により検索されるタプルがどのようにデータページに分散されるかの情報を索引情報530中に含めておき、それを利用してアクセスされるデータページ数をより詳細に求めてもよい。

【0 1 0 0】

この処理の全部又は一部として、SQL解析モジュール252は、DBMS90で実行計画

570を作成するとき内部的に見積られた値を実行計画570と同時に出力させ、その値を利用してもよい。求めた値を対応するエントリ292に設定する。SQL解析モジュール252は、これらの処理をステップ2503で把握した全ての繰り返しグループに関して実施する(ステップ2505)。

【0 1 0 1】

最後に、SQL解析モジュール252は、シーケンシャルヒントの設定を行う。まず、SQL解析詳細情報のエントリ288を参照して、その方法が“全走査” “BitMap索引へのアクセス” “BitMap索引を利用した表へのアクセス” であるものに対応するシーケンシャルヒントのエントリ294の値を“Y” に設定する。続いて、SQL解析モジュール252は、先読みJob情報350中の入力関連情報のエントリ430を参照し、その中に登録されているデータIDがエントリ286に登録されている駆動データIDと一致し、データ構造名が一致するエントリに対応するシーケンシャルヒントのエントリ294の値を“Y” に設定する。

【0 1 0 2】

その後、SQL解析モジュール252は、シーケンシャルヒントのエントリ294に“Y” が設定されたデータを駆動データとしてネストループ結合が行われるデータが存在するかを、既已取得した実行計画570から把握する。そのようなデータが存在する場合には、先読みJob情報350中の表データ順情報のエントリ420を参照して、駆動データと結合データのデータ順を調べ、同様のデータ順である場合には、結合データに関しても対応するシーケンシャルヒントのエントリ294の値を“Y” に設定する(ステップ2506)。その後、SQL解析モジュール252は処理を完了する(ステップ2507)。

【0 1 0 3】

上述した処理により、事前の情報収集処理が行われる。

【0 1 0 4】

以下、Jobプログラム100が実行される際の、先読みプログラム160による先読み指示処理に関して説明する。

【0 1 0 5】

図16は、先読み指示処理の処理手順を示す図である。本処理は、先読み方法

決定モジュール254が、Job状態情報800としてJobプログラム100の開始をJob管理プログラム130から受け取ることにより開始され、Job状態情報800としてJobプログラム100の完了を受信した後に後処理が実施され、処理が完了する。なお、Job状態情報800は、常に状態が示されるJobプログラム100の識別子であるプログラムIDと共に送られる。また、Jobプログラム100の開始を示すJob状態情報800は、必要に応じてキャッシュ量情報を含む(ステップ1101)。

【0106】

次に、先読み方法決定モジュール254は、繰り返し情報805として入力データ量をJob管理プログラム130から受信する。この入力データ量は、Jobプログラム100に対して入力として与えられるデータの件数であり、入力データのデータIDとその件数を表すデータの組として与えられる。本実施形態においては、入力データは、これから実施されるJobプログラム100の前に実施された他のJobプログラム100の出力データを利用する。その件数を先に実施されたJobプログラム100にデータ量810としてJob管理プログラム130に対して出力させ、その値を元にこれから実施するJobプログラム100のデータ件数をJob管理プログラム130が計算し、繰り返し情報805の入力データ量として与える。なお、本ステップは必ずしも実施される必要はない(ステップ1102)。

【0107】

次に、先読み方法決定モジュール254は、ステップ1102で取得した入力データ量、Job状態情報800中のキャッシュ量情報、SQL解析情報280中のSQL解析詳細情報290からDBMS90に対して指示するキャッシュ量設定710と先読み方法720を決定する。

【0108】

図17は、先読み方法決定モジュール254がDBMS90に対して指示するキャッシュ量設定710のデータ構造を示す図である。キャッシュ量設定710には、キャッシュ量の設定が指示されるデータ構造のデータ構造名を保持するエントリ711及び最低限利用すべきキャッシュ量を保持するエントリ712の組が含まれる。複数のDBMS90が関与する場合には、先読み方法決定モジュール254は、これらのエントリをDBMS90毎に用意する。

【0 1 0 9】

図 1 8 は、先読みプログラム160内で利用される、先読み方法720のデータ構造を示す図である。先読み方法720には、先読みやキャッシュ指示を行うデータ構造のデータ構造名を保持するエントリ721、先読み方法・キャッシュ方法を保持するエントリ722、対応する記憶装置40の装置IDを登録するエントリ723、その記憶装置40で利用したいデータキャッシュ42の割当量を示すキャッシュ量が登録されるエントリ724及びそのデータへのアクセス順を保持するエントリ725の組が含まれる。なお、複数のDBMS90が関与する場合には、更に先読み方法720にDBMS IDを保持するエントリが追加される。

【0 1 1 0】

まず、先読み方法決定モジュール254は、SQL解析情報280から、処理開始時に与えられたプログラムIDに対応するSQL解析詳細情報290を選択する。その中で、エントリ288に登録されたアクセス方法が“全走査”のデータ構造に対しては、記憶装置40及びDBMS90の双方に、“全走査”アクセス向けにそれぞれ独立に事前に定められた一定量のキャッシュを割り当てるとする。次に、エントリ288に登録されたアクセス方法が“全走査”ではなく、かつ、シーケンシャルヒントのエントリ292の値が“Y”であるデータ構造に対しては、先読み方法決定モジュール254は、全走査の場合よりも多い、それぞれ独立に事前に定められたキャッシュ量を記憶装置40・DBMS90の双方に割り当てる。そして、先読み方法決定モジュール254は、これらのデータ構造に関しては、エントリ722に“シーケンシャル”を登録する。

【0 1 1 1】

それ以外のデータ構造に関しては、先読み方法決定モジュール254は、まず、処理実行を保証するため、事前に定められた最低キャッシュ量を、記憶装置40・DBMS90の双方に割り当て、残りのキャッシュを以下の方法でそれらのデータ構造に分配していく。

【0 1 1 2】

全ての駆動データIDに関してステップ1102でそれに一致するデータIDを持つ入力データ量が与えられ、かつ、これからキャッシュ量を設定すべき全てのデータ

構造で対応する期待アクセスページ数のエントリ292に値が保持されている場合には、先読み方法決定モジュール254は、 $(\text{駆動データIDに対応する入力データ量}) \times (\text{期待アクセスページ数}) / (\text{そのデータ構造のデータページ数})$ を指標として、その値が大きなデータ構造から順に、 $(\text{駆動データIDに対応する入力データ量}) \times (\text{期待アクセスページ数}) \times (\text{データページサイズ}) \times (\text{事前設定割合})$ 、又は $(\text{データ構造のデータページ数}) \times (\text{データページサイズ}) \times (\text{事前設定割合})$ のうち値の小さな方に対応する量をそのデータ構造に割り当てる。

【0113】

その後、先読み方法決定モジュール254は、割り当てられるキャッシュ量の総和が、DBMS90毎にエントリ340で与えられたキャッシュ量の値になるまで繰り返す。その後、先読み方法決定モジュール254は、同じ指標を用いて、引き続いて記憶装置40に、記憶装置40毎にエントリ340で与えられるキャッシュ量の値になるまでキャッシュの割当を繰り返す。

【0114】

前記の条件を満たさず、前記の指標を計算できないデータ構造が存在する場合には、先読み方法決定モジュール254は、キャッシュ割当の優先順位決定の指標として (データ構造のデータページ数) を、キャッシュ割当量として $(\text{データ構造のデータページ数}) \times (\text{データページサイズ}) \times (\text{事前設定割合})$ を用いて上述した処理と同様の処理を行う。これらの方法で記憶装置40にキャッシュが割り当てられたデータ構造に対しては、先読み方法決定モジュール254はエントリ722に、“即時先読み”を登録する。

【0115】

データ構造のデータページに関する情報は、システム情報300中の索引情報530から、対応するエントリを参照することにより求めることができる。また、記憶装置40毎のキャッシュ量を求める必要があるが、先読み方法決定モジュール254は、システム情報300中のデータ記憶領域情報510と領域マッピング情報310を参照し、データ構造がどの記憶装置40に記憶されているか求める。あるデータ構造が複数の記憶装置40に記憶される場合には、先読み方法決定モジュール254は、原則としてそれぞれのデータ量に比例してキャッシュ量を記憶装置40に分配する

。ただし、いずれかの記憶装置40でエントリ340に登録されたキャッシュ量の制約を超える場合には、先読み方法決定モジュール254は、キャッシュ量の制約分までその記憶装置40にキャッシュ量を割り当てた後に、残りの記憶装置40間でそれぞれのデータ量に比例してキャッシュ量を分配する。

【0 1 1 6】

上記の方法で求めたキャッシュ割当に従って、先読み方法決定モジュール254は、キャッシュ量設定710と先読み方法720へ値を設定する。なお、先読み方法720中のエントリ725には、SQL解析詳細情報290中の対応する値がそのまま設定される。

【0 1 1 7】

なお、必ずしもキャッシュ量情報340が与えられるとは限らない。その場合には、先読み方法決定モジュール254は、DBMS90や記憶装置40で利用可能なキャッシュ量は、それぞれの全キャッシュ量の、事前設定される割合分であると判断する。

【0 1 1 8】

また、先読み方法決定モジュール254がDBMS90と記憶装置40の双方のキャッシュ量の設定を行うと説明したが、DBMS90のキャッシュ量は固定とし、記憶装置40のキャッシュ量のみ割当を動的に変更してもよい。この場合には、先読み方法決定モジュール254は、前述のDBMS90に対してキャッシュ割当を行う際と同じ指標でデータ構造へのキャッシュ割当優先順位とキャッシュ割当量を求める。そして、先読み方法決定モジュール254は、優先度が高いデータ構造から順に、現在のDBMS90におけるそのデータ構造において最低限利用可能なキャッシュ量の、求めたキャッシュ割当量からの不足分を、記憶装置40のキャッシュを利用するとして、記憶装置40のキャッシュの割当を行う。先読み方法決定モジュール254は、上述の処理を記憶装置40で割当可能なキャッシュ量が0になるまで繰り返す(ステップ1105)。

【0 1 1 9】

先読み方法決定モジュール254は、ステップ1105で求めたキャッシュ量設定710を対応するDBMS90に対して指示する。なお、先読み方法決定モジュール254は、D

BMS90に対して指示する前に、DBMS90の設定前のキャッシュ量の設定を取得し、別途保存しておく。この指示を基に（必要なら独自の判断を加えながら）、DBMS90はキャッシュ量の設定の変更を行う。なお、DBMS90のキャッシュ量を固定として変更しない場合は、本ステップは実施しない(ステップ1106)。

【0 1 2 0】

次に、先読み方法決定モジュール254は、ステップ1105で求めた先読み方法720を先読み指示モジュール256に与え、記憶装置40に対してキャッシュ指示730を発行するように要求する。

【0 1 2 1】

図19は、キャッシュ指示730のデータ構造を示す図である。キャッシュ指示730には、複数の領域を1つにまとめるための識別子であるグループ化を保持するエントリ732、記憶装置40におけるデータ領域を示す、LU等の仮想構造の識別子及びその領域を示す情報からなるデータ領域を保持するエントリ734、キャッシュ方法を保持するエントリ735、キャッシュ量を保持するエントリ736、並びにアクセス順を保持するエントリ737の組が含まれる。

【0 1 2 2】

依頼を受けた先読み指示モジュール256は、先読み方法720中のデータ構造名と装置IDから、システム情報300中のデータ記憶領域情報510と領域マッピング情報310を用いて、それぞれの記憶装置40におけるデータ領域を判別し、記憶装置40毎にキャッシュ指示730を作成する。ここで、エントリ735、736及び737に関しては、先読み方法720に登録されたキャッシュ方法、キャッシュ量及びアクセス順に対応する値がそのまま設定される。グループ化に関しては、同じデータ構造名と装置IDの組に対応するが記憶装置上では非連続なデータ領域に分割される際に同一の値が設定され、それ以外のものは異なる値が設定される。

【0 1 2 3】

その後、先読み指示モジュール256は、作成したキャッシュ指示730に対応する記憶装置40に対して送信する。キャッシュ指示730を受け取った記憶装置40の制御プログラム44は、その指示に従って、データキャッシュ42の管理や先読み処理を実施する。

【 0 1 2 4 】

なお、先読み方法決定モジュール254は、先読み指示モジュール256に依頼した先読み方法720を別途記憶しておく(ステップ1107)。

【 0 1 2 5 】

その後、先読み方法決定モジュール254は、Job管理プログラム130から、Job状態情報800としてJobプログラム100の完了報告を受信するまで処理を一時停止する(ステップ1108)。

【 0 1 2 6 】

Job状態情報800として処理の完了報告を受信した後、先読み方法決定モジュール254は、DBMS90や記憶装置40に、設定したキャッシュの設定の解除指示を発行する。具体的には、先読み方法決定モジュール254は、ステップ1106でDBMS90にキャッシュ量の変更を指示した場合、そのステップで保存された指示前のキャッシュ設定に戻すキャッシュ量設定710をDBMS90に送信する。この指示を基に、DBMS90はキャッシュ量の設定を元に戻す。

【 0 1 2 7 】

また、先読み方法決定モジュール254は、先読み指示モジュール256に対して、保存しておいた先読み情報720内のエントリ722を全て“設定解除”にし、エントリ724の値を全て0に設定した先読み情報720を送信し、キャッシュ指示730の発行を先読み指示モジュール256に依頼する。先読み指示モジュール256は、与えられた先読み情報720を基に、ステップ1107と同様に対応する記憶装置40に対してキャッシュ指示730を発行し、キャッシュ設定の解除を指示する。キャッシュ指示730を受け取った記憶装置40の制御プログラム44は、その指示に従って、データキャッシュ42の管理を元の状態に戻し、前に与えられたキャッシュ指示730に従ったデータの先読みを終了する(ステップ1109)。

【 0 1 2 8 】

これにより、全ての処理が完了する(ステップ1120)。

【 0 1 2 9 】

ここまで、記憶装置40はLU208を外部装置に提供し、外部装置はI/Oパス34を経由してLU208にアクセスすると説明してきた。しかし、本発明は、記憶装置40が

ファイル202を外部装置に提供し、そのファイル202がネットワークファイルシステムプロトコルを用いてネットワーク24経由でアクセスされる構成にも適用できる。

【0 1 3 0】

図20は、記憶装置40がファイル202を外部装置に提供する計算機システムの構成を示す図である。この場合、本図の計算機システムは、図1と比較して、以下の点で違いがある。

【0 1 3 1】

I/Oパス34や仮想化スイッチ60が存在しない。サーバ70はI/OパスI/F 32を有さない。OS72は、外部装置が提供するファイル202をネットワークファイルシステムプロトコルを用いてネットワークI/F 22及びネットワーク24経由でアクセスするネットワークファイルシステム82を有し、ボリュームマネージャ78やファイルシステム80を有さなくともよい。ネットワークファイルシステム82は、領域マッピング情報310をOS管理情報74中に有する。DBMS90により認識されるファイル202と記憶装置40から提供されるファイル202があるルールに従って対応する場合、その対応関係を定めるルールの情報のみがOS管理情報74中に保持されても良い。この場合には、先読みプログラム160は対応関係を定める情報を取得し、それから領域マッピング情報310を作成し、システム情報300中に記憶する。

【0 1 3 2】

記憶装置40はI/OパスI/F 32を有さなくても良く、ファイルを外部装置に対して提供する。記憶装置40の制御プログラム44は、図1のファイルシステム80と同等のプログラムを有し、記憶装置40内に存在するLU208の記憶領域を仮想化し、ファイル202として提供する。また、制御プログラム44は1つ以上のネットワークファイルシステムプロトコルを解釈し、ネットワーク24及びネットワークI/F 22経由で外部装置からそのプロトコルを用いて要求されるファイルアクセスを処理する。この記憶装置40では、キャッシュ指示730は、エントリ734にはファイルの識別子とその領域を示す情報が登録され、ファイル202をベースに外部からのキャッシュ領域やそのキャッシュ方法が指示されることができる。

【0 1 3 3】

データのマッピングに関しては、図2で説明したデータのマッピング階層構成において、ファイル202以下が全て記憶装置40により提供されるようになり、サーバ70は、OS72内のネットワークファイルシステム82を用いて記憶装置40上にあるファイル202をアクセスする。

【0134】

記憶装置40がファイル202を外部装置に提供する場合、上述した各処理においては、LU208に対応する部分が記憶装置40上のファイル202に置き換えられる。

【0135】

次に、本発明の第二の実施形態について説明する。第二の実施形態では、繰り返し実行されるSQL文を処理開始時に先読みプログラムが取得し、その解析結果を基に先読み指示を発行する。尚、第二の実施形態は第一の実施形態と同一の部分が多い。以下、第二の実施形態と第一の実施形態が異なる部分のみ説明し、同一な部分については説明を省略する。尚、第二の実施形態における計算機システムの構成や各機器が保持するデータのデータ構造は、以下に述べる個所以外、原則として第一の実施形態と同一である。

【0136】

図21は、第二の実施形態における、先読み処理に関する先読みプログラム160やその他プログラムと、それらプログラムが保持する、もしくは、それらプログラム間で交換される情報を示したブロック図である。先読みプログラム160は、Job管理プログラム130から繰り返し情報805を受け取る代わりに、Jobプログラム100から繰り返し情報805bを受け取る。また、抽出SQL情報820をJobプログラム100が実施される前に取得したのに代わり、先読みプログラム160は、ストアドプロシージャ情報840をJobプログラム100の実行前に、Jobプログラム100が実施されるときにJobプログラム100からSQLヒント830を受け取る。なお、本図では、Job状態情報800をJob管理プログラムから受け取っているが、これをJobプログラム100から受け取るようにすることもできる。

【0137】

図22は、先読みプログラム160が事前に実施する情報収集処理の処理手順を示す図である。

【 0 1 3 8 】

ステップ2102、ステップ2103及びステップ2106については、ステップ2101からの処理の場合と同一の処理を行う。

【 0 1 3 9 】

ステップ2103の処理の終了後、SQL解析モジュール252は、先読みJob情報350で指定されるJobプログラム100が発行するSQL文のうち、ストアードプロシージャ化されるSQL文をストアードプロシージャ情報840として取得する。

【 0 1 4 0 】

図 2 3 は、ストアードプロシージャ情報840に含まれるストアードプロシージャの宣言の例5200を示す。この例5200では、範囲5202がストアードプロシージャの呼び出し名を示す。ストアードプロシージャ情報840は、開発プログラム150中のストアードプロシージャ把握モジュール272が開発コード152を元に作成する。具体的には、ストアードプロシージャ把握モジュール272が、開発コード152に含まれるソースコードに含まれるSQL文を解析し、ストアードプロシージャの宣言部分を把握し、それを抽出することにより作成する。

【 0 1 4 1 】

複数のストアードプロシージャが利用される場合には、それらを全て抽出してストアードプロシージャ情報840が作成される。なお、プログラムIDを指定してストアードプロシージャ情報840の作成を開発プログラム150に依頼し、それを先読みプログラム160に与える処理は管理者が行っても良いし、先読みプログラム160中のSQL解析モジュール252が直接行っても良い（ステップ2104b）。

【 0 1 4 2 】

SQL解析モジュール252は、取得したストアードプロシージャ情報840に含まれるストアードプロシージャをそれぞれ分離し、分離された各々のストアードプロシージャに関してSQL解析詳細情報290bをそれぞれ独立に作成する。

【 0 1 4 3 】

図 2 4 は、SQL解析詳細情報290bのデータ構造を示す図である。SQL解析詳細情報290との違いは、繰り返しグループID、実行順、及び駆動データIDを保持するエントリの替わりに、解析されたSQL文である被解析SQL文を保持するエントリ29

6及びストアドプロシージャの呼び出し名であるストアドプロシージャ名を保持するエントリ298が付加されることである。

【0 1 4 4】

SQL解析詳細情報290bの作成方法は、ステップ2501から始まるSQL解析詳細情報290を作成する処理とほぼ同一である。ただし、本実施形態では、本ステップでは、1つのストアドプロシージャは第一の実施形態における繰り返しグループに相当するとして扱い、繰り返しグループに対応して設定した繰り返しグループID、実行順、及び駆動データIDの設定処理が行われない。

【0 1 4 5】

また、SQL解析モジュール252は、ストアドプロシージャの宣言を被解析SQL文のエントリ296に設定し、その宣言を解析して得られるストアドプロシージャの呼び出し名をエントリ298に設定する（ステップ2105b）。

【0 1 4 6】

尚、本実施形態では、Jobプログラム100が繰り返し情報805bやSQLヒント830を発行する必要がある。ここで、繰り返し情報805bは、繰り返し処理の開始もしくは終了を示す情報で、繰り返し処理の開始を示す場合には、必要に応じて処理の繰り返し回数も含まれる。SQLヒント830は、これから実行される繰り返し処理構造の中で実行される一連のSQL文700である。なお、繰り返し情報805bやSQLヒント830は常にJobプログラム100のプログラムIDと共に送信されるようにし、その送信元Jobプログラム100のプログラムIDが識別できるようにする。

【0 1 4 7】

図25は、C言語で書かれたソースコードに埋め込みSQL文が含まれている場合に、そのソースコードに、Jobプログラム100から繰り返し情報805bやSQLヒント830を発行させるための埋め込み文を追加する処理による変換例を示す図である。この処理は、開発プログラム150中のSQLヒント埋め込みモジュール274により行われる。

【0 1 4 8】

ソースコードの範囲5002で示される部分では、for文により繰り返し処理が行われ、その中で幾つかのSQL文が実行される。SQLヒント埋め込みモジュール274

はこの繰り返し構造を識別し、その中にSQL文が存在していることから、そのSQL文が繰り返し実行されると判断する。この場合、SQLヒント埋め込みモジュール274は、その繰り返し構造が開始される直前に、繰り返し処理の開始を先読みプログラム160に伝える繰り返し情報805bをJobプログラム100に発行させるための埋め込み文5022と、SQLヒント830を先読みプログラム160へ発行するための埋め込み文5026を挿入する。また、SQLヒント埋め込みモジュール274は、その繰り返し構造が終了した直後に、繰り返し処理が完了を先読みプログラム160に伝える繰り返し情報805bをJobプログラム100に発行させるための埋め込み文5028を挿入する。

【0149】

ここで、埋め込み文5022には、繰り返し回数を示す変数の値を出力するように、出力変数を示す情報5024が付加されてもよい。また、埋め込み文5026から出力されるSQLヒント830は、範囲5002中の埋め込みSQL文を抜き出した情報5010である。

【0150】

ソースコードは、このヒント出力を行う埋め込み文が追加された後に、更に実行形式を作成する処理が行われ、こうして作成された実行形式がJobプログラム100として実行される。

【0151】

図26は、ソースコードがSQLスクリプトを用いて記述された場合に、そのSQLスクリプトを解釈・実行するスクリプト実行プログラムを用いてJobプログラム100として実行する時に、繰り返し情報805bやSQLヒント830の発行をスクリプト実行プログラムに指示する文をSQLスクリプトに追加する処理例を示す図である。この処理もSQLヒント埋め込みモジュール274により行われる。本例のSQLスクリプトでは、範囲5102でカーソルの定義が行われ、範囲5104でカーソルを用いて読み出されたデータ毎に、範囲5106の処理が繰り返し実行される。

【0152】

SQLヒント埋め込みモジュール274は、この繰り返し構造を識別し、繰り返し処理が実施されるされる範囲5104の直前に、繰り返し処理の開始を先読みプログラ

ム160に伝える繰り返し情報805bの発行をスクリプト実行プログラムに指示する文5022b及びSQLヒント830の先読みプログラム160への発行をスクリプト実行プログラムに指示する文5026bを挿入する。また、その繰り返し構造が終了した直後に、SQLヒント埋め込みモジュール274は、繰り返し処理の完了を先読みプログラム160に伝える繰り返し情報805bの発行をスクリプト実行プログラムに指示する文5028を挿入する。ここで、文5022bには、繰り返し回数を示す変数の値を出力するように繰り返し数をカウントする文5024bが付加されてもよい。また、埋め込み文5026bから出力されるSQLヒント830は、範囲5104中の実際に繰り返し実行されるSQL文を抜き出した情報5110である。

【0 1 5 3】

Jobプログラム100実行時に、この変換されたSQLスクリプトがスクリプト実行プログラムに与えられ、繰り返し情報805bやSQLヒント830を出力させながら処理が実施される。また、この解析機能をスクリプト実行プログラムに持たせ、SQLスクリプトの実行時に動的に繰り返し情報805bやSQLヒント830の生成・発行が行われても良い。

【0 1 5 4】

以下、本実施形態における、Jobプログラム100が実行される際の、先読みプログラム160による先読み指示処理に関して説明する。

【0 1 5 5】

図 2 7 は、本実施形態における先読み指示処理の手順を示す図である。なお、本実施形態においては、先読みプログラム160がJob状態情報800としてJobプログラム100の開始をJob管理プログラム130から受け取ることにより本処理が開始され、先読みプログラム160がJob状態情報800としてJobプログラム100の完了を受信したときに処理が完了する。なお、前述のように、Job状態情報800はJobプログラム100が送信してもよい（ステップ1101b）。

【0 1 5 6】

まず、先読み処理プログラム160の先読み方法決定モジュール254は、繰り返し情報805bとSQLヒント830をJobプログラム100から受信する。なお、繰り返し情報805b中に、繰り返し回数が与えられても与えられなくてもよい（ステップ1103b

）。

【0 1 5 7】

続いて、先読み方法決定モジュール254は、SQLヒント830中からSQL文700を把握し、それをSQL解析モジュール252に与えてSQL解析詳細情報290bを作成させ、SQL解析情報280中に保存させる。なお、ここで作成されたSQL解析詳細情報290bには、ストアプロシージャ名を保持すべきエントリ298に値は設定されない。また、SQL文700の中にストアプロシージャを呼び出す部分が存在する場合には、その部分の解析結果として、そのストアプロシージャに対応して作成されたSQL解析詳細情報290bの情報がそのまま用いられる。

【0 1 5 8】

また、先読み方法決定モジュール254は、本ステップでは、SQLヒント830で与えられた全体が、第一の実施形態における1つの繰り返しグループに相当すると判断する。それ以外のSQL解析詳細情報290bの設定は、ステップ2105bで述べた方法と同様にして行われる（ステップ1104b）。

【0 1 5 9】

続いて、先読み方法決定モジュール254及び先読み指示モジュール256は、ステップ1105bからステップ1107bまでの処理を行う。これらの処理は、第一の実施形態のステップ1105からステップ1107で説明した処理と同様であるが、以下のような違いが存在する。

【0 1 6 0】

まず、利用されるSQL解析詳細情報290bは、ステップ1104bで作成したものである。また、SQL解析詳細情報290bにはアクセス順が登録されるエントリが存在せず、先読み方法720及びキャッシュ指示730では、アクセス順を保持するエントリを抹消するか、もしくは、無効値や同一の値を保持させるようにする。

【0 1 6 1】

続いて、先読み方法決定モジュール254は、Jobプログラム100が発行する繰り返し情報805bとしての繰り返し処理の完了報告を受信するまで処理を一時停止する（ステップ1108b）

その後、先読み方法決定モジュール254は、DBMS90や記憶装置40に設定したキ

キャッシュの設定の解除指示を発行する。それら処理の詳細は、第一の実施形態で説明したステップ1109と同様である（ステップ1109b）。

【0 1 6 2】

その後、先読み方法決定モジュール254は、Jobプログラム100からの情報、もしくは、Job状態情報800の受信待ち状態に入る。Job状態情報800としてのJobプログラム100の完了報告を受信した場合には、先読み方法決定モジュール254は処理を完了する（ステップ1120b）。それ以外の情報を受信した場合には、ステップ1103bに戻り、受信した情報の確認をする（ステップ1110b）。

【0 1 6 3】

尚、本実施形態も、記憶装置40がファイル202を外部装置に提供し、そのファイル202がネットワークファイルシステムプロトコルを用いてネットワーク24経由でアクセスされる計算機システムにも適用可能である。そのときの注意点は、第一の実施形態と同様である。

【0 1 6 4】

次に、本発明の第三の実施形態について説明する。第三の実施形態では、先読みプログラム160はDBMS90のフロントエンドプログラムの振舞う。先読みプログラム160は、与えられたSQL文が繰り返し実行されるとして解析後、先読み指示を発行し、その後にDBMS90に対してSQL文を転送する。第三の実施形態は、第二の実施の形態と同一の部分が多い。以下、第三の実施形態と第二の実施形態が異なる部分のみ説明し、同一な部分については説明を省略する。尚、第三の実施形態における計算機システムの構成や各機器が保持するデータのデータ構造は、以下に述べる個所以外、原則として第二の実施形態と同一である。

【0 1 6 5】

図28は、第三の実施形態における、先読み処理に関する先読みプログラム160やその他プログラムと、それらプログラムが保持する、もしくは、それらプログラム間で交換される情報を示したブロック図を示す図である。先読みプログラム160は、Jobプログラム100実行時にSQLヒント830を受け取る代わりに、処理要求として最終的にDBMS90に対して送信されるSQL文700を受領し、それを用いて必要な処理を実行後、そのSQL文700をDBMS90に対して送信する。その処理結果と

して、実行結果950をDBMS90から受領し、それをそのままJobプログラム100に対して返信する。なお、本図では、Job状態情報800をJob管理プログラムから受け取っているが、第二の実施の形態と同様に、これをJobプログラム100から受け取るようにすることもできる。

【0 1 6 6】

先読みプログラム160が事前に実施する情報収集処理の処理に関しては、第二の実施形態と同一で、ステップ2101bから開始される処理を行う。

【0 1 6 7】

本実施形態では、Jobプログラム100が繰り返し情報805bを発行する必要がある。以下、その方法について説明する。

【0 1 6 8】

図 2 9 は、C言語で書かれたソースコードに埋め込みSQL文が含まれている場合に、繰り返し情報805bをJobプログラム100に発行させるための埋め込み文を追加する処理による変換例を示す図である。この処理は、開発プログラム150中の繰り返し情報埋め込みモジュール276により行われる。本処理は、第二の実施形態におけるSQLヒント埋め込みモジュール274による変換とほぼ同一であるが、繰り返し情報埋め込みモジュール276の場合には、SQLヒント830をJobプログラム100に発行させるための埋め込み文5026を挿入しない点が異なる。

【0 1 6 9】

図 3 0 は、ソースコードがSQLスクリプトを用いて記述された場合に、そのSQLスクリプトを解釈・実行するスクリプト実行プログラムを用いてJobプログラム100として実行する時に、繰り返し情報805bの発行をスクリプト実行プログラムに指示する文を追加する処理による変換例を示す図である。この処理も繰り返し情報埋め込みモジュール276により行われる。本処理もSQLヒント埋め込みモジュール274による変換とほぼ同一であるが、繰り返し情報埋め込みモジュール276の場合には、SQLヒント830の発行をスクリプト実行プログラムに指示する文5026bを挿入しない点が異なる。

【0 1 7 0】

Jobプログラム100実行時に、この変換されたSQLスクリプトがスクリプト実行

プログラムに与えられ、繰り返し情報805bを出力させながら処理が実施される。また、この解析機能をスクリプト実行プログラムに持たせ、SQLスクリプトの実行時に動的に繰り返し情報805bの生成・発行が行われても良い。

【 0 1 7 1 】

以下、本実施形態における、Jobプログラム100が実行される際の、先読みプログラム160による先読み指示処理に関して説明する。図31は、本実施形態における先読み指示処理の手順を示す図である。なお、本実施形態においては、先読み方法決定モジュール254がJob状態情報800としてJobプログラム100の開始をJob管理プログラム130から受け取ることにより本処理が開始され、先読み方法決定モジュール254がJob状態情報800としてJobプログラム100の完了を受信したときに処理が完了する。なお、前述のように、Job状態情報800はJobプログラム100が送信してもよい（ステップ1201）。

【 0 1 7 2 】

まず、先読み方法決定モジュール254は、繰り返し情報805bをJobプログラム100から受信する。なお、繰り返し情報805b中に、繰り返し回数を与えられても与えられなくてもよい（ステップ1202）。

【 0 1 7 3 】

続いて、先読み方法決定モジュール254は、DBMS90に対して処理要求として発行されるSQL文700をJobプログラム100から受信する。なお、このSQL文700は、送信元Jobプログラム100のプログラムIDと共に送信する等、送信元のJobプログラム100のプログラムIDが識別可能であるようにする（ステップ1203）。

【 0 1 7 4 】

続いて、先読み方法決定モジュール254は、ステップ1203で受信したSQL文700に対応するSQL解析詳細情報290bがSQL解析情報280中に存在するか確認する（ステップ1204）。存在する場合にはステップ1209に進み、存在しない場合にはステップ1205に進む。

【 0 1 7 5 】

SQL解析詳細情報290bがSQL解析情報280中に存在しない場合、先読み方法決定モジュール254はSQL解析モジュールに対し、ステップ1203で受信したSQL文700に

ついて、SQL解析詳細情報290bを作成し、SQL解析情報280中に保存するよう指示する。その作成方法は、ステップ1104bと同様である（ステップ1205）。

【 0 1 7 6 】

続いて、先読み方法決定モジュール254及び先読み指示モジュール256は、ステップ1105cからステップ1107cまでの処理を行う。これらの処理は、第二の実施形態で説明したステップ1105bからステップ1107bと同様であるが、以下のような違いが存在する。第二の実施形態における処理では、あるJobプログラム100に対応するSQL解析詳細情報290bが増えることはないが、本処理では、対応するSQL解析詳細情報290bが順次増加する。

【 0 1 7 7 】

また、ステップ1105cでキャッシュ量設定710や先読み方法720を決定する際には、先読み方法決定モジュール254は、それまでに発行したものは特に考慮せずに新たに最適と思われるキャッシュ量設定710や先読み方法720を順次決定する。また、ステップ1106cで、指示する前のDBMS90の設定を保存する場合には、本処理を開始する前のDBMS90の設定が常に保存される。さらに、ステップ1107cで、先読み方法720が記憶されるが、それは最後に先読み指示モジュール256に依頼された先読み方法720が記憶される。

【 0 1 7 8 】

ステップ1207cの実行後、又はステップ1204でSQL解析詳細情報290bがSQL解析情報280中に存在すると判断された場合、先読み方法決定モジュール254は、ステップ1203で受信したSQL文700を対応するDBMS90に対して発行し、その処理結果を取得する。その後、先読み方法決定モジュール254は、SQL文700を発行したJobプログラム100に対して、取得した処理結果をそのまま返す（ステップ1209）。

【 0 1 7 9 】

続いて、先読み方法決定モジュール254は、Jobプログラム100からの情報受信待ち状態に入り、Jobプログラム100からの繰り返し情報805bとしての繰り返し処理の完了報告の受信の有無を確認する。受信した情報が完了報告以外の情報の場合には、先読み方法決定モジュール254は、ステップ1203に戻り、受信した情報の確認をする（ステップ1210）。

【0180】

繰り返し情報805bとしての繰り返し完了処理の報告を受信した場合、先読み方法決定モジュール254は、第二の実施形態のステップ1109bと同様の処理を行う（ステップ1211）。

【0181】

その後、先読み方法決定モジュール254は、Jobプログラム100からの情報、もしくは、Job状態情報800の受信待ち状態に入る。情報を受信したときに、それがJob状態情報800としてのJobプログラム100の完了報告であるか確認する（ステップ1212）。

【0182】

受信した情報がJob状態情報800としてのJobプログラム100の完了報告でない場合には、先読み方法決定モジュール254は、ステップ1202に戻り、受信した情報の確認をする。

【0183】

受信した情報がJob状態情報800としてのJobプログラム100の完了報告である場合には、先読み方法決定モジュール254は、処理が完了したJobプログラム100に対応する、ストアドプロシージャの解析結果でない、つまり、ストアドプロシージャ名を保持するエントリ298に値を持たないSQL解析詳細情報290bをSQL解析情報280から削除する。なお、Jobプログラム100との対応関係はプログラムIDを用いて把握される（ステップ1213）。そして、処理を完了する（ステップ1214）。

【0184】

尚、本実施形態も、記憶装置40がファイル202を外部装置に提供し、そのファイル202がネットワークファイルシステムプロトコルを用いてネットワーク24経由でアクセスされる計算機システムにも適用可能である。そのときの注意点は、第一の実施形態と同様である。

【0185】**【発明の効果】**

本発明により、DBMSが稼動する計算機システムにおいて、同じ形のSQL文で与えられる処理が多数回、繰り返し実行される場合に、記憶装置へのアクセス性能

が向上する。

【図面の簡単な説明】

【図 1】

第一の実施の形態における計算機システムの構成を示す図である。

【図 2】

第一の実施の形態におけるデータマッピングの階層構成の概念を示す図である。

【図 3】

領域マッピング情報310のデータ構造を示す図である。

【図 4】

データ記憶領域情報510のデータ構造を示す図である。

【図 5】

表データ量情報520のデータ構造を示す図である。

【図 6】

索引情報530のデータ構造を示す図である。

【図 7】

Job実行管理情報360のデータ構造を示す図である。

【図 8】

第一の実施の形態における、先読み処理に関係する先読みプログラム160やその他プログラム間で交換される情報の流れを示す図である。

【図 9】

第一の実施の形態における、先読みプログラム160が事前に実施する情報収集処理の手順を示す図である。

【図 1 0】

先読みJob情報350のデータ構造を示す図である。

【図 1 1】

第一の実施の形態における抽出処理の例を示す図である。

【図 1 2】

第一の実施の形態における抽出処理の例を示す図である。

【図 1 3】

抽出SQL情報820からSQL解析詳細情報290を作成する処理の手順を示す図である。

【図 1 4】

SQL解析詳細情報290のデータ構造を示す図である。

【図 1 5】

実行計画570のデータ構造を示す図である。

【図 1 6】

第一の実施の形態における、先読みプログラム160による先読み指示処理の手順を示す図である。

【図 1 7】

キャッシュ量設定710のデータ構造を示す図である。

【図 1 8】

先読み方法720のデータ構造を示す図である。

【図 1 9】

キャッシュ指示730のデータ構造を示す図である。

【図 2 0】

第一の実施の形態において、記憶装置40がファイル202を外部装置に提供する場合の計算機システムの構成を示す図である。

【図 2 1】

第二の実施の形態における、先読み処理に関する先読みプログラム160やその他プログラム間で交換される情報の流れを示す図である。

【図 2 2】

第二の実施の形態における、先読みプログラム160が事前に実施する情報収集処理の手順を示す図である。

【図 2 3】

ストアドプロシージャの宣言の例を示す図である。

【図 2 4】

SQL解析詳細情報290bのデータ構造を示す図である。

【図 2 5】

第二の実施の形態における変換例を示す図である。

【図 2 6】

第二の実施の形態における変換例を示す図である。

【図 2 7】

第二の実施の形態における、先読みプログラム160による先読み指示処理の手順を示す図である。

【図 2 8】

第三の実施の形態における、先読み処理に関する先読みプログラム160やその他プログラム間で交換される情報の流れを示す図である。

【図 2 9】

第三の実施の形態における変換例を示す図である。

【図 3 0】

第二の実施の形態における変換例を示す図である。

【図 3 1】

第三の実施の形態における、先読みプログラム160による先読み指示処理の手順を示す図である。

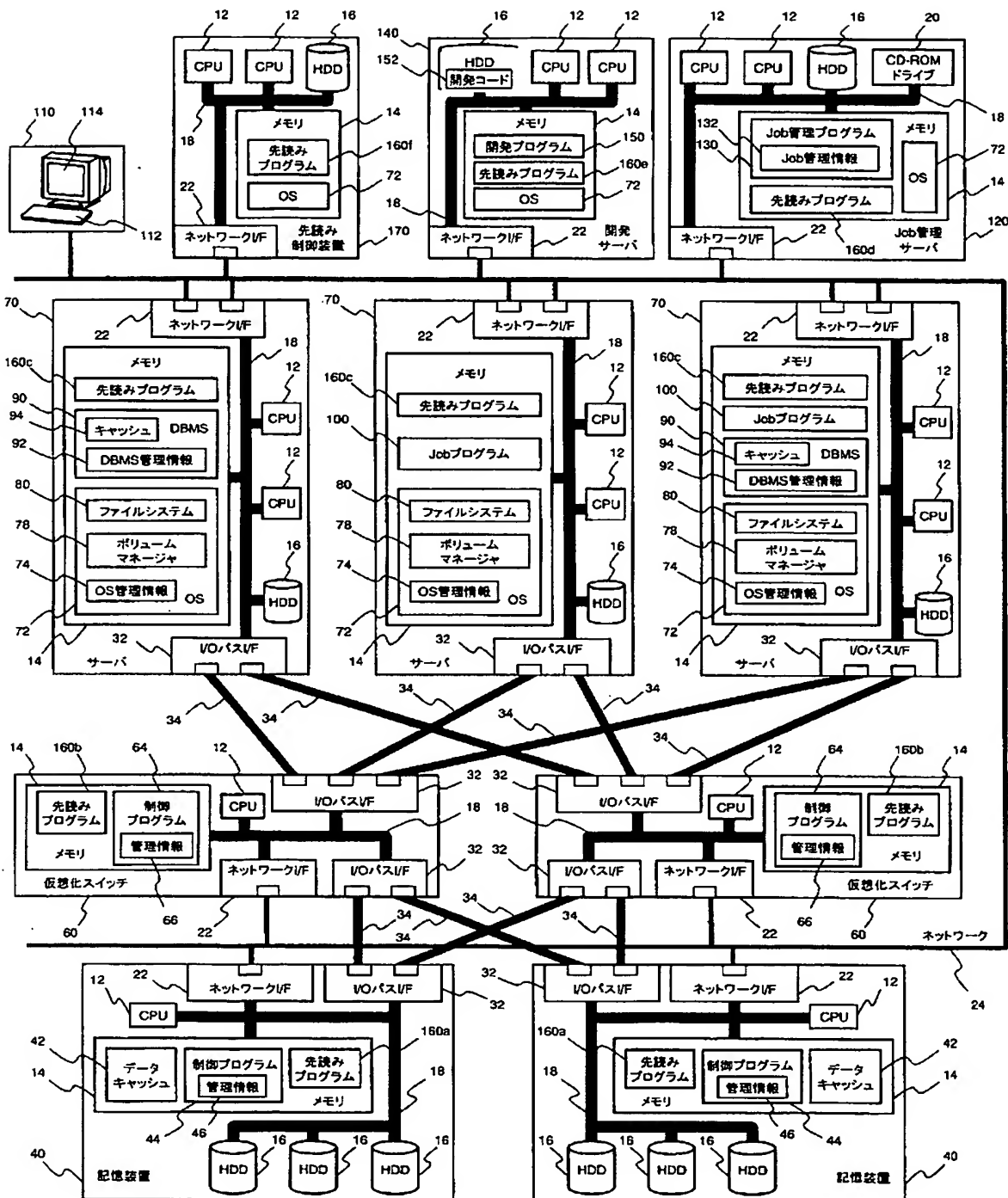
【符号の説明】

16…HDD、22…ネットワークI/F、24…ネットワーク、32…I/OパスI/F、34…I/Oパス、40…記憶装置、60…仮想化スイッチ、70…サーバ、90…DBMS、100…Jobプログラム、130…Job管理プログラム、150…開発プログラム、160、160a、160b、160c、160d、160e、160f…先読みプログラム

【書類名】 図面

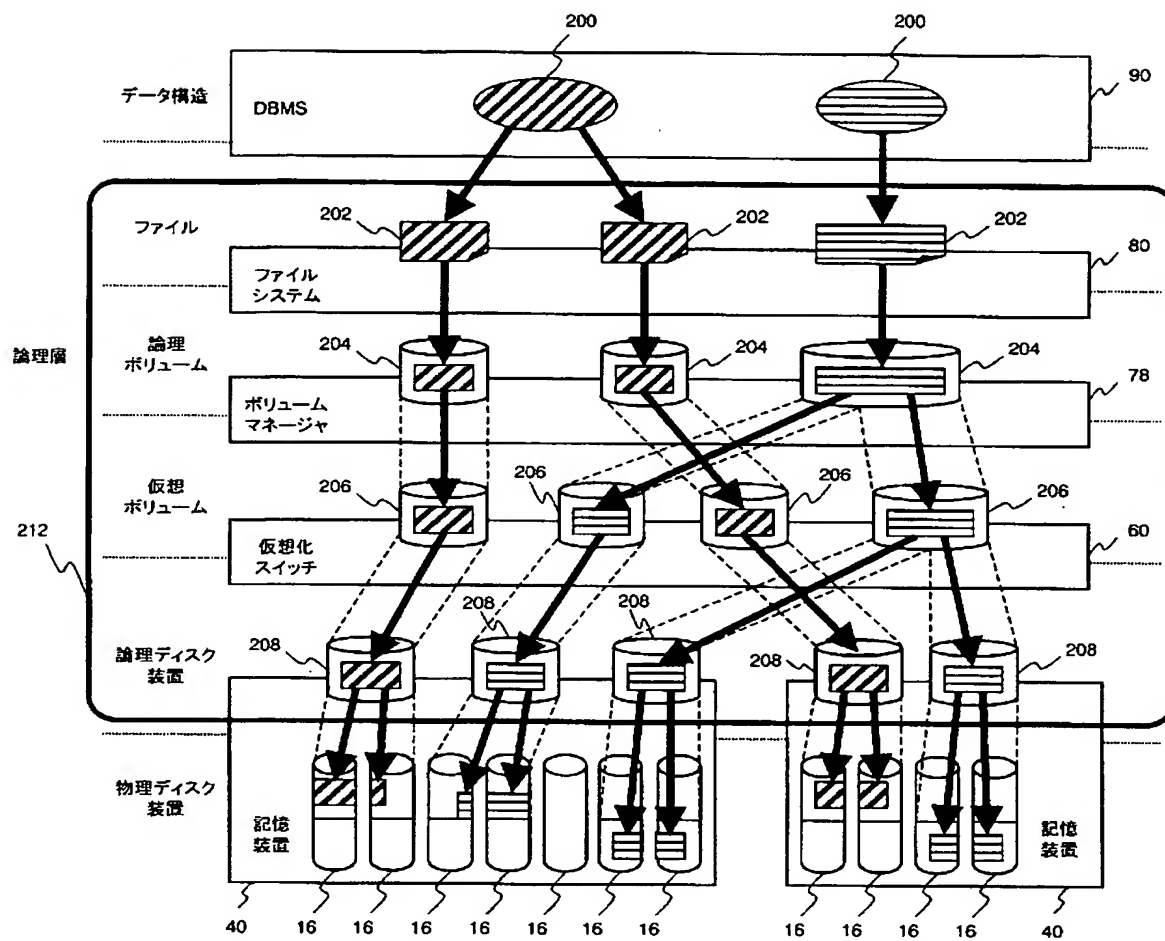
【図 1】

図 1



【図 2】

图2



【図 3】

図3

上位仮想構造 領域情報		下位管理構造 対応領域情報		
仮想構造ID	構造内領域	仮想化機構ID	管理構造ID	構造内領域
Upper 0	0 - 10239	LowApp0	Lower 0	0 - 10239
	10240 - 20479	LowApp1	Lower 1	0 - 10239
Upper 1	0 - 10239	LowApp0	Lower 0	10240 - 20479
	10240 - 20479	LowApp1	Lower 1	10240 - 20479
Upper 2	0 - 10239	LowApp2	Lower 0	0 - 10239
	10240 - 20479	LowApp2	Lower 1	0 - 10239
⋮	⋮	⋮	⋮	⋮

領域マッピング情報

【図 4】

図4

データ 構造名	データ記憶位置	
	データファイルパス名	ブロック番号
T1	/dev/rdisk/lvol0	0 - 499
	/dev/rdisk/lvol0	2000 - 2499
T2	/dev/rdisk/lvol0	500 - 799
T3	/dev/rdisk/lvol1	500 - 799
⋮	⋮	⋮

データ記憶領域情報

【図5】

図5

データ構造名	T1	T2	...	521
データページサイズ	16kB	8kB	...	522
データページ数	61897	78783	...	524
キャッシュ量	1024ページ	2048ページ	...	526
表データ量情報				520

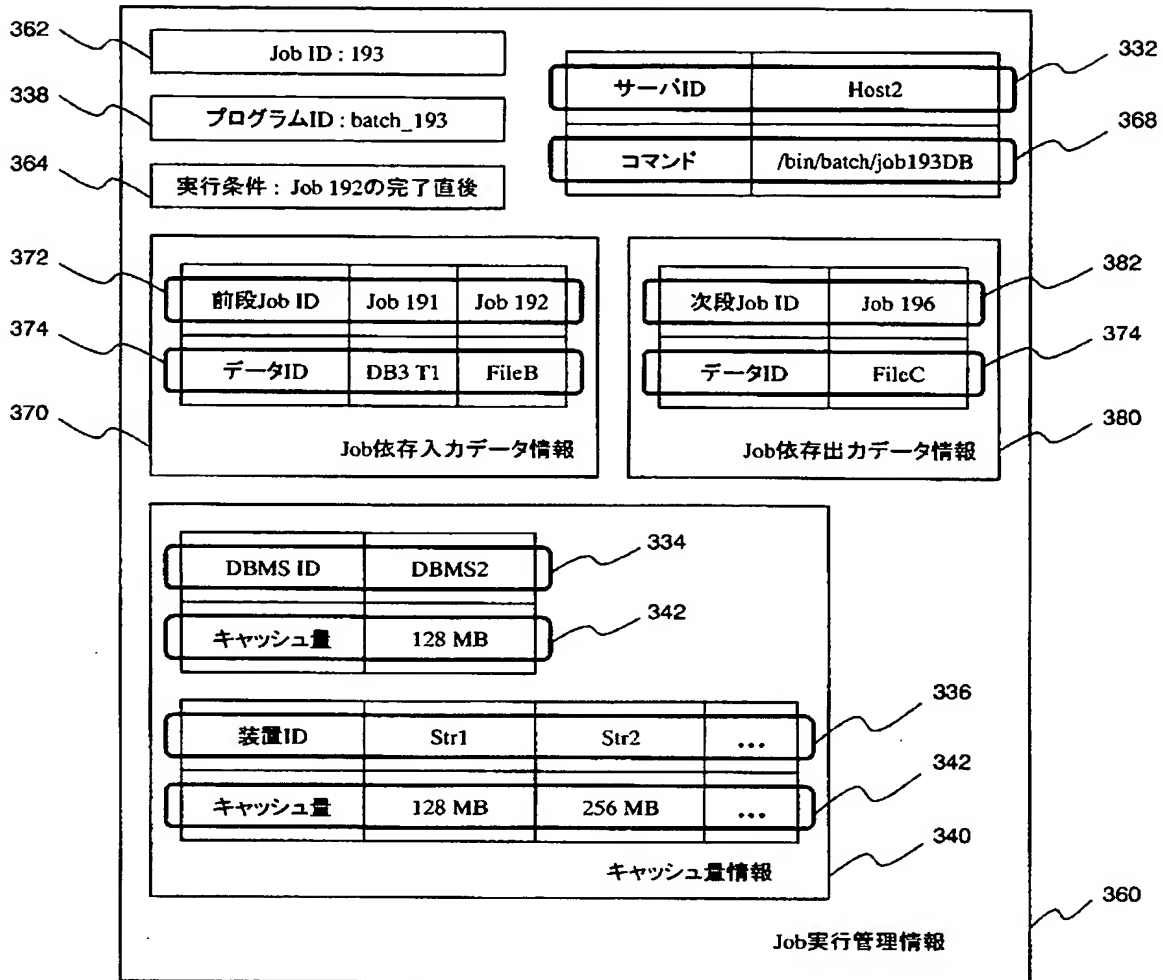
【図6】

図6

データ構造名	Ind1-1	Ind2-1	...	531
対応表名	T1	T2	...	532
索引種別	B-Tree	BitMap	...	534
データページサイズ	8kB	8kB	...	533
データページ数	3782	6782	...	535
Leafノードページ数	2898	-	...	536
キャッシュ量	1452ページ	50ページ	...	537
検索属性	ID1	(ID1,ID2)	ID2	538
期待タプル数	5.5	1	2058.2	542
索引情報				530

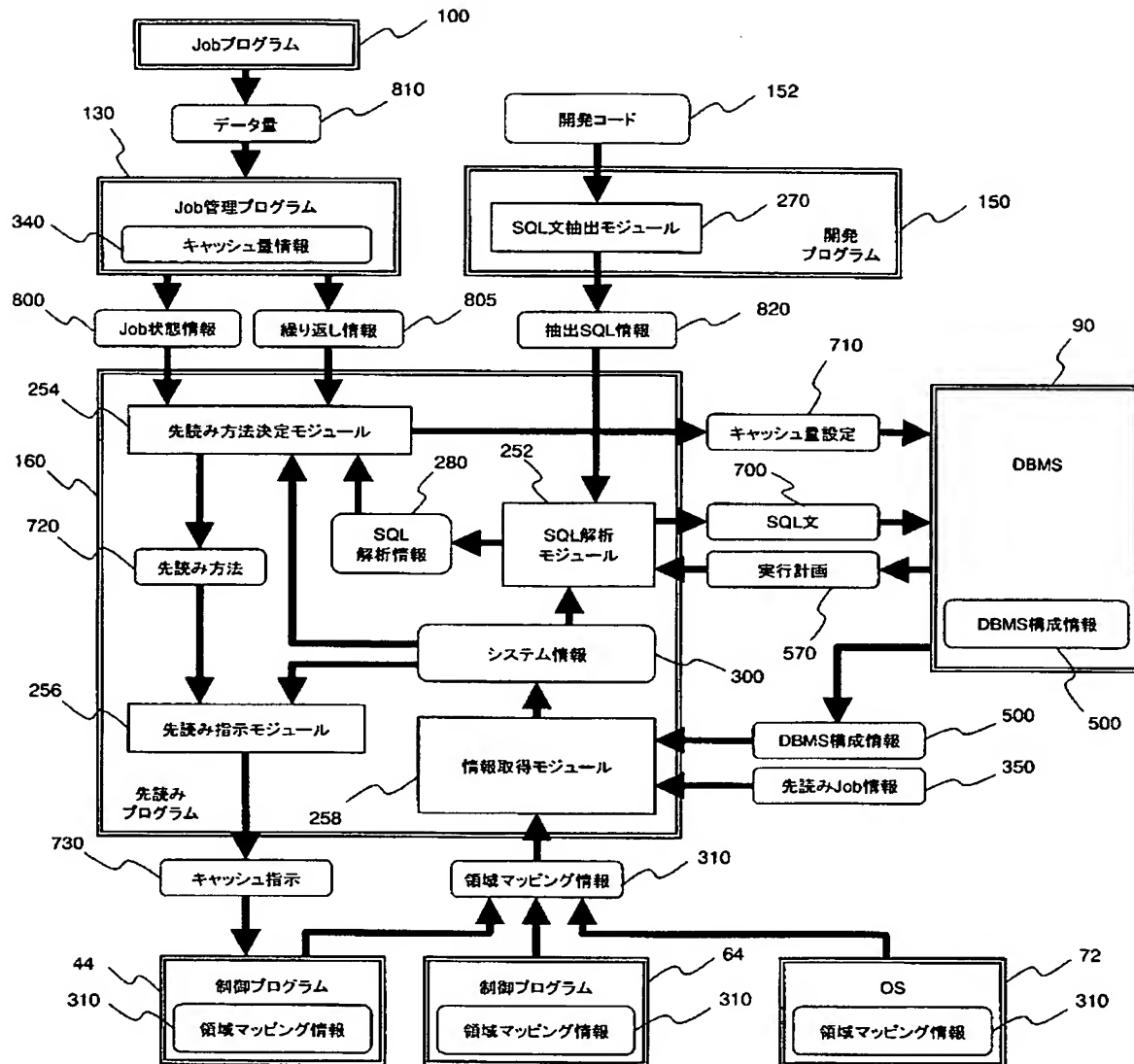
【図 7】

図7



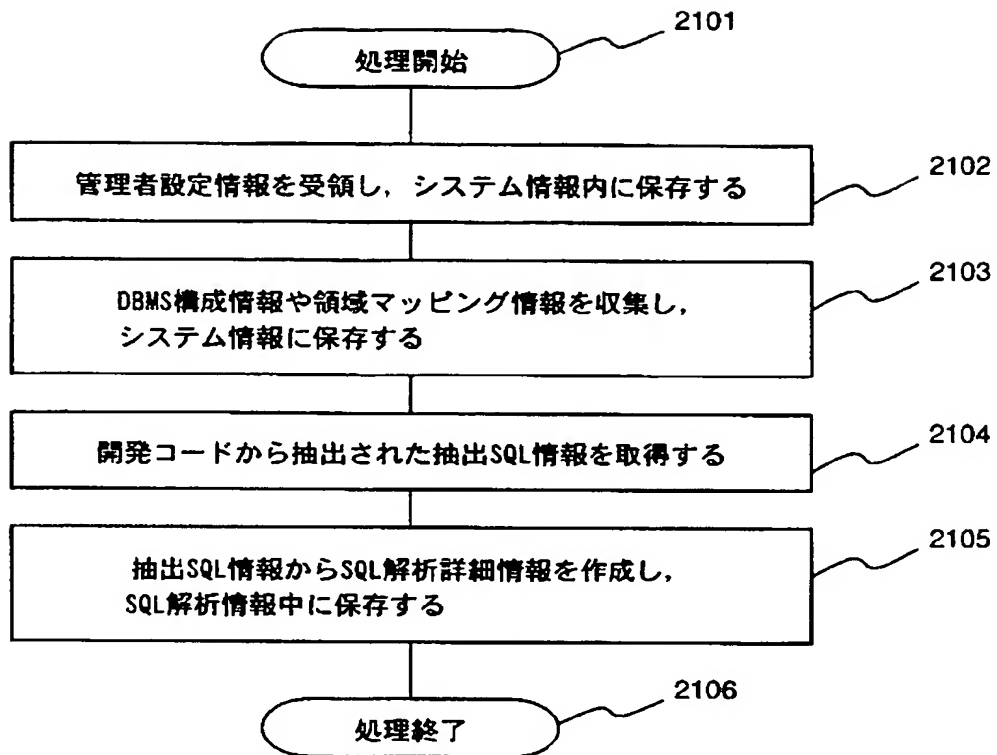
【図 8】

図8



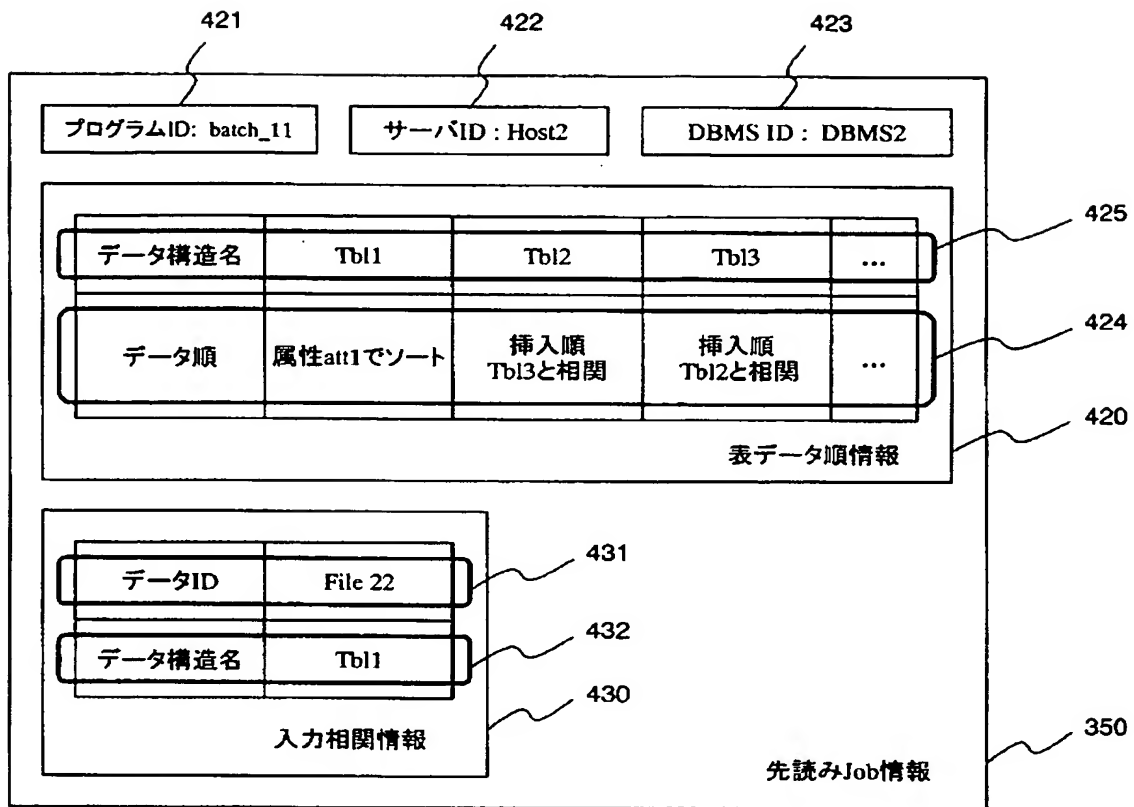
【図9】

図9



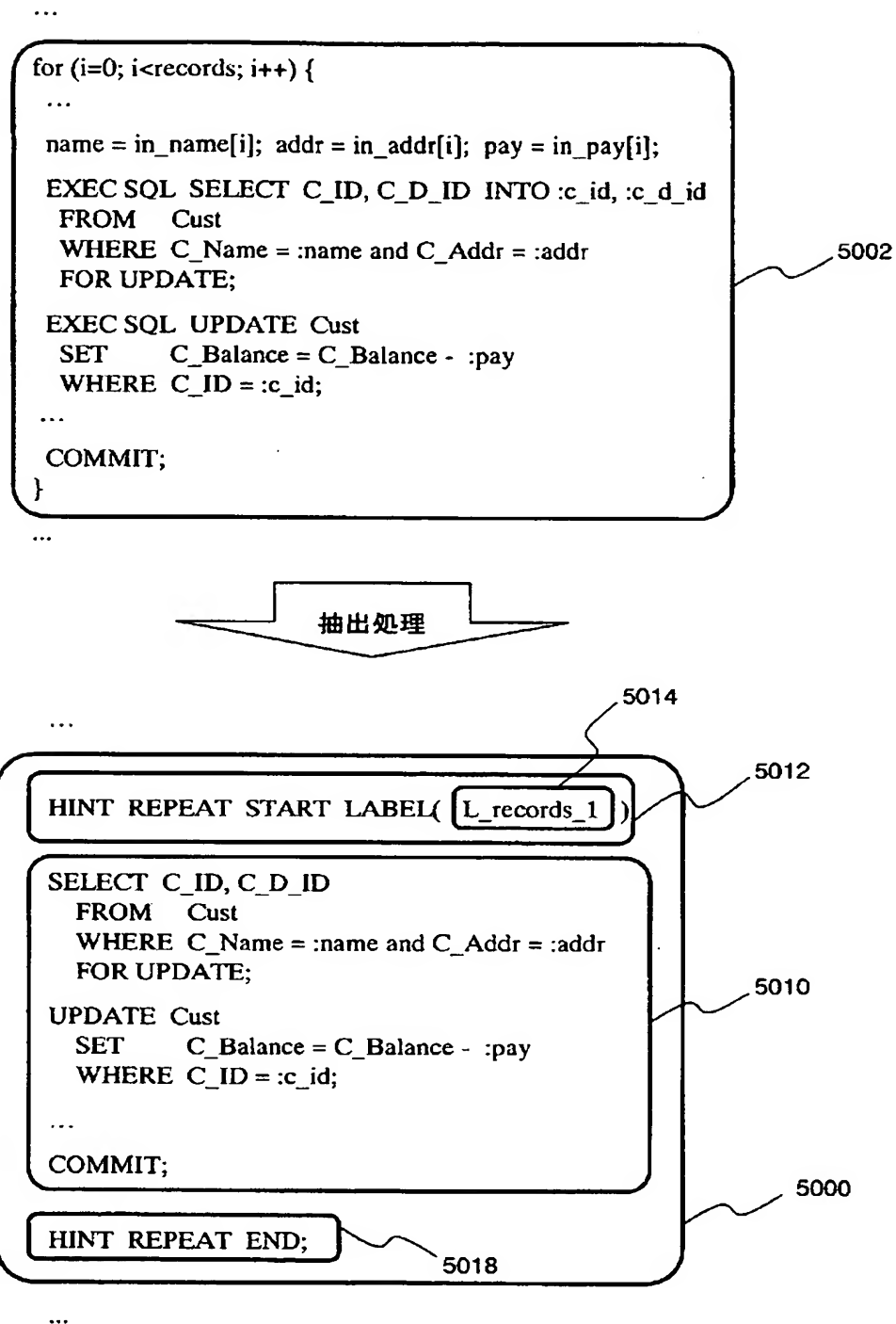
【図10】

図10



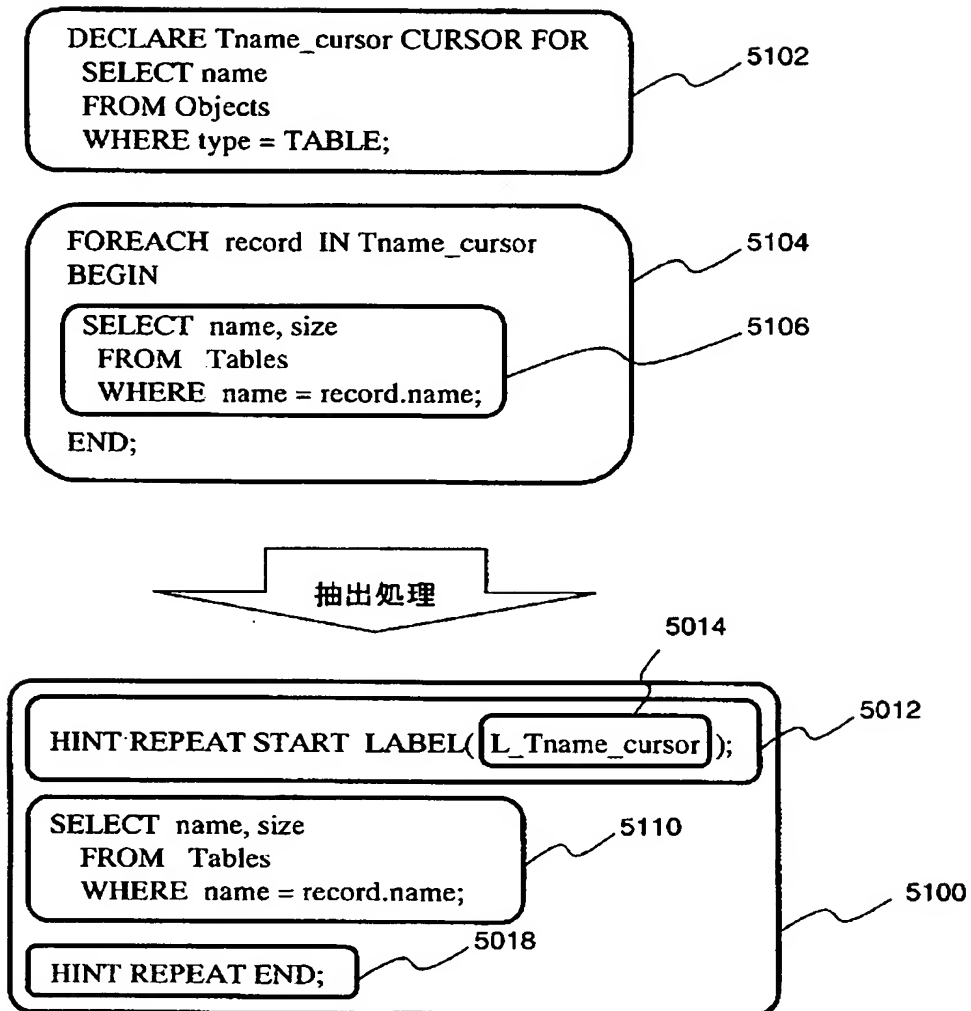
【図 11】

図 11



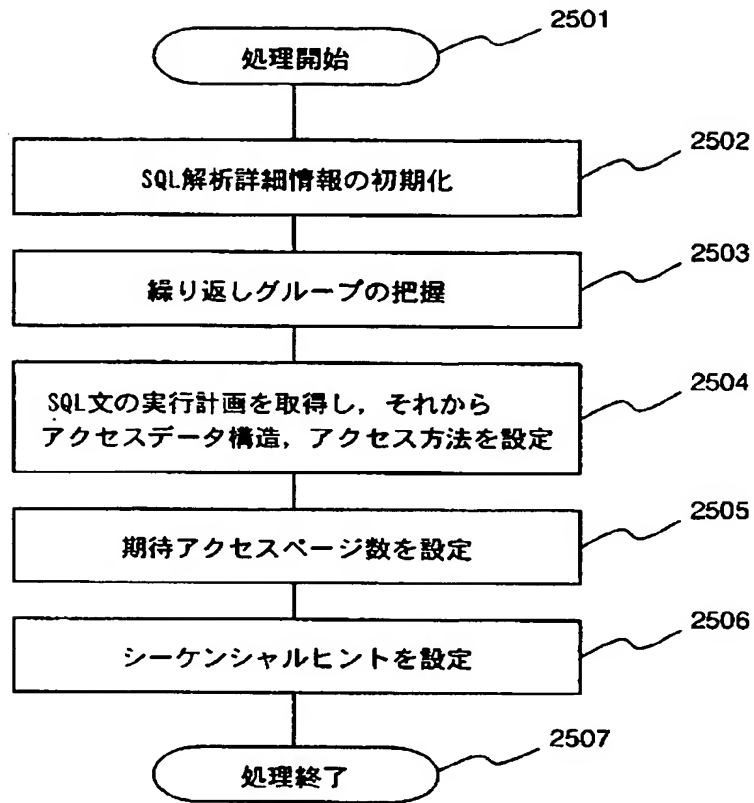
【図 12】

図 12



【図 13】

図13



【図 14】

図 14

プログラムID: batch_11		DBMS ID: DBMS3	
繰り返しグループID	L_loop_1	...	
実行順	1	...	
駆動データID	DB8 T1	...	
データ構造名	T1	Ind2-1	...
アクセス方法	Table Access Full	Index Access	...
期待アクセスページ数	-	4.5	...
シーケンシャルヒント	Y	-	...

SQL解析詳細情報

【図 15】

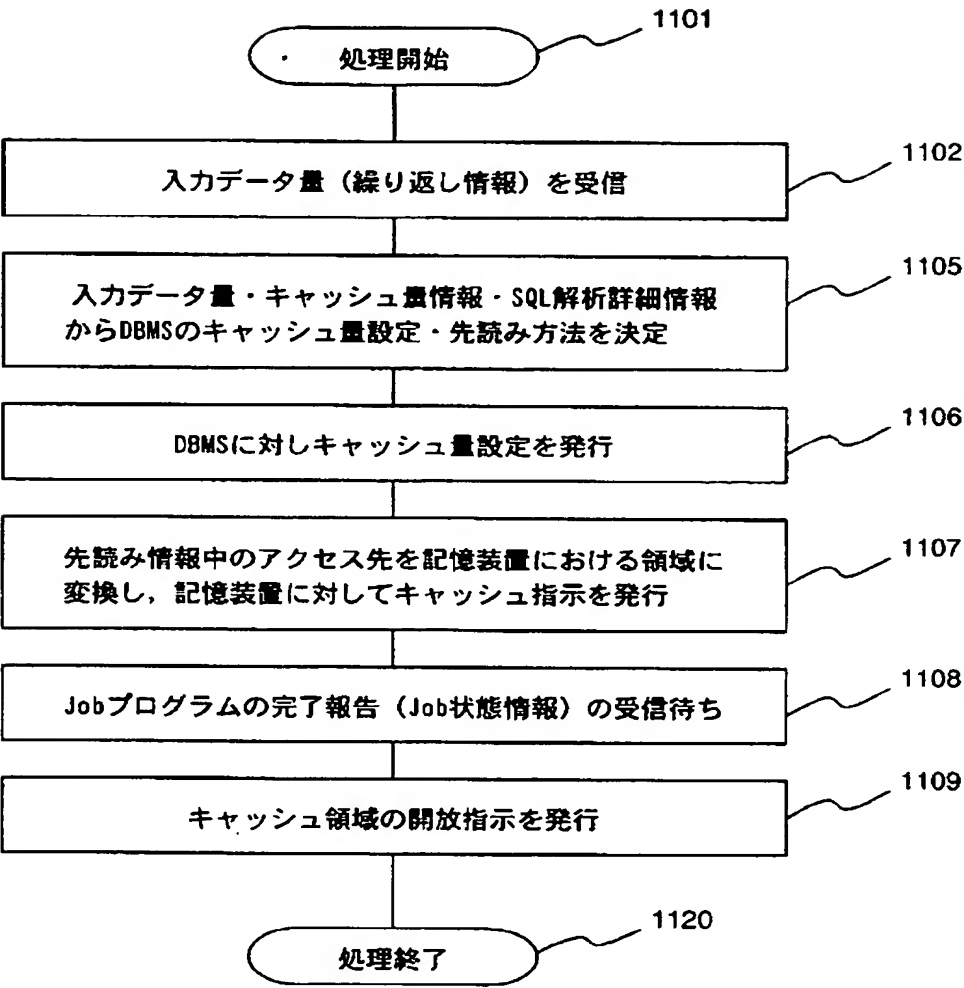
図 15

ノード名	親ノード名	ノード処理内容	アクセスデータ構造	ノード処理詳細
N-1-1	root	Sum	-	Sum(T1.B)
N-2-1	N-1-1	Nested Loop Join	-	T1.Key1 = T3.Key1
N-3-1	N-2-1	Hash Join	-	T3.Key3 = T4.Key4
N-3-2	N-2-1	Table Access by Index	T1	-
N-4-1	N-3-1	Table Access Full	T3	-
N-4-2	N-3-1	Filter	-	T4.M < 100
N-4-3	N-3-2	Index Access	Ind 1-1	Key1=[N-3-1の結果]
N-5-1	N-4-2	Table Access Full	T4	-

実行計画

【図 16】

図 16



【図 17】

図 17

データ構造名	T1	T2	...
キャッシュ量	1024ページ	2048ページ	...

キャッシュ量設定

Figure 17 shows a table structure for cache amount setting. The table has two rows and four columns. The first row is labeled 'データ構造名' (Data structure name) and contains 'T1', 'T2', and '...'. The second row is labeled 'キャッシュ量' (Cache amount) and contains '1024ページ' (1024 pages), '2048ページ' (2048 pages), and '...'. The table is enclosed in a box labeled 'キャッシュ量設定' (Cache amount setting). Reference numerals 710, 711, and 712 point to the table structure.

【図18】

図18

データ構造名	T1	Ind2-1	...	721
キャッシュ方法	シーケンシャル	即時先読み	...	722
装置ID	Str2	Str1	...	723
キャッシュ量	4MB	40MB	...	724
アクセス順	1	2	...	725
先読み方法				720

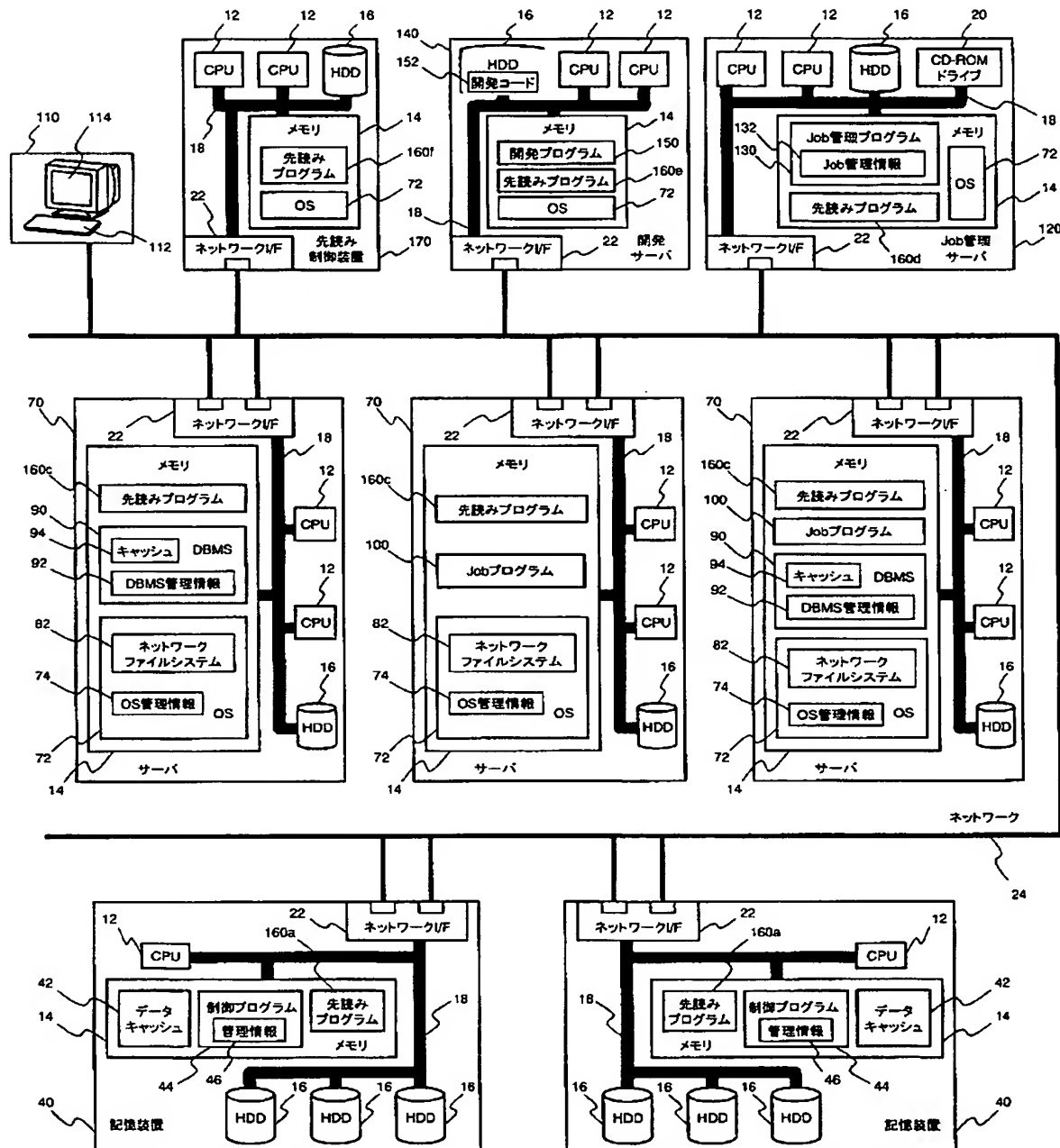
【図19】

図19

グループ化		1	2	...	732
データ領域	仮想構造識別子	LU 1	LU 2	...	734
	領域	10240- 819199	81920 - 163839	...	
キャッシュ方法		シーケンシャル	即時先読み	...	735
キャッシュ量		4MB	40MB	...	736
アクセス順		1	2	...	737
キャッシュ指示					730

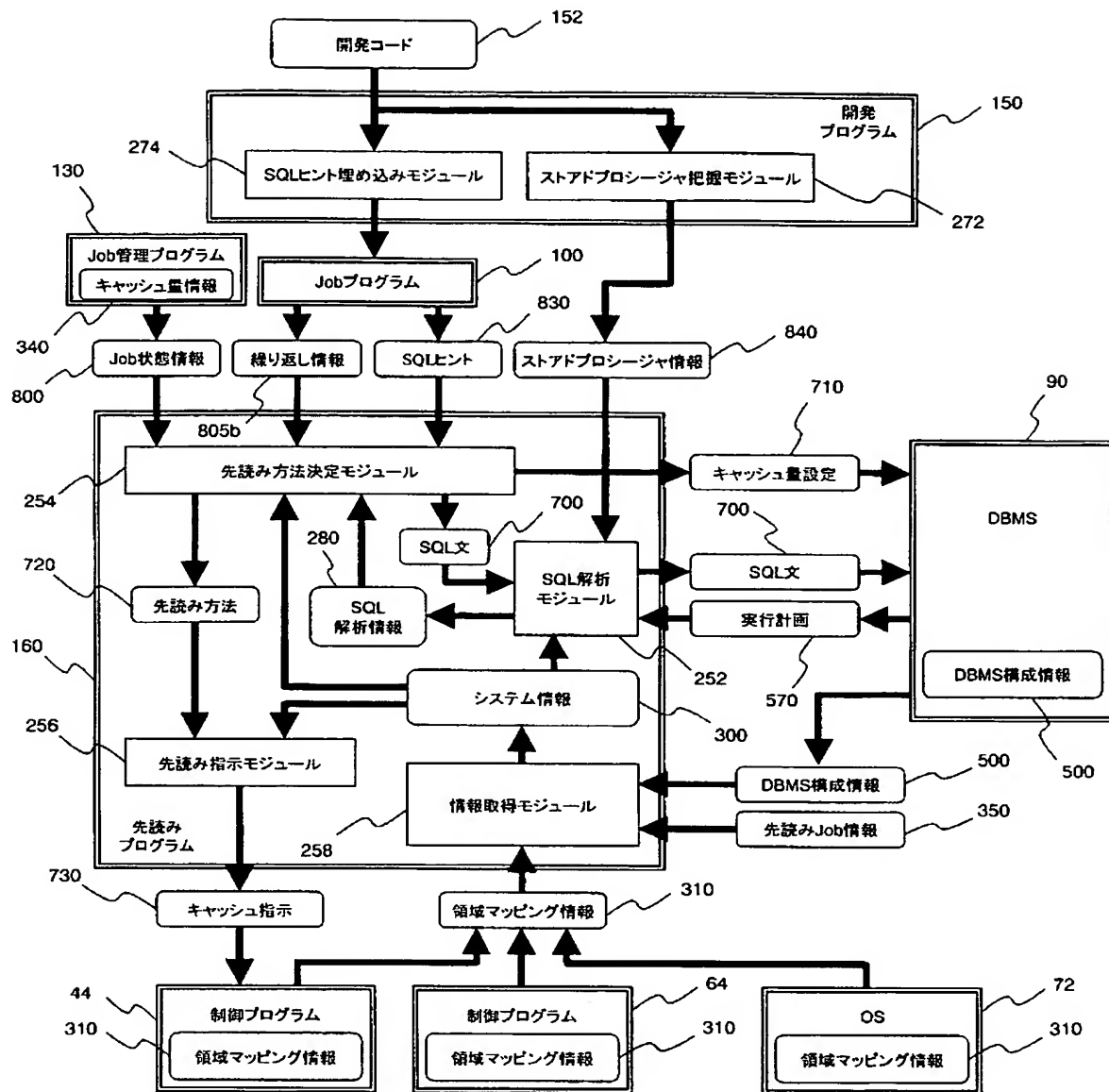
【図 20】

図20



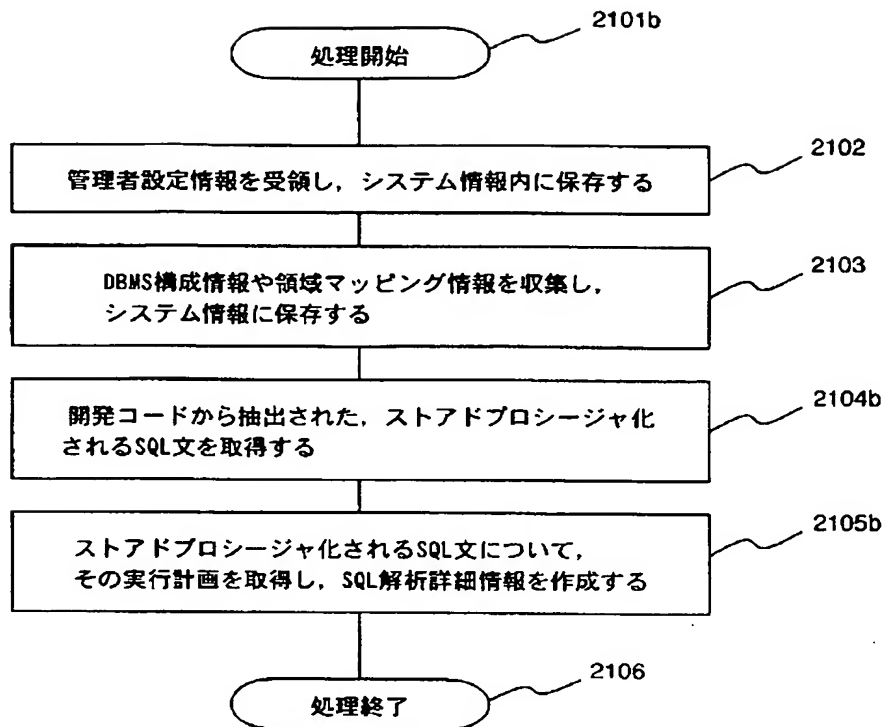
【図 21】

図21



【図 22】

図22



【図 23】

図23

5202

```
CREATE PROCEDURE proc_pay (IN name, IN address, IN pay)
BEGIN
  DECLARE c_id   NUMBER(10);
  DECLARE c_d_id NUMBER(10);

  SELECT C_ID, C_D_ID INTO c_id, d_id
  FROM   Cust
  WHERE  C_Name = name and C_Addr = addr
  FOR UPDATE;

  UPDATE Cust
  SET    C_Balance = C_Balance - pay
  WHERE  C_ID =c_id;

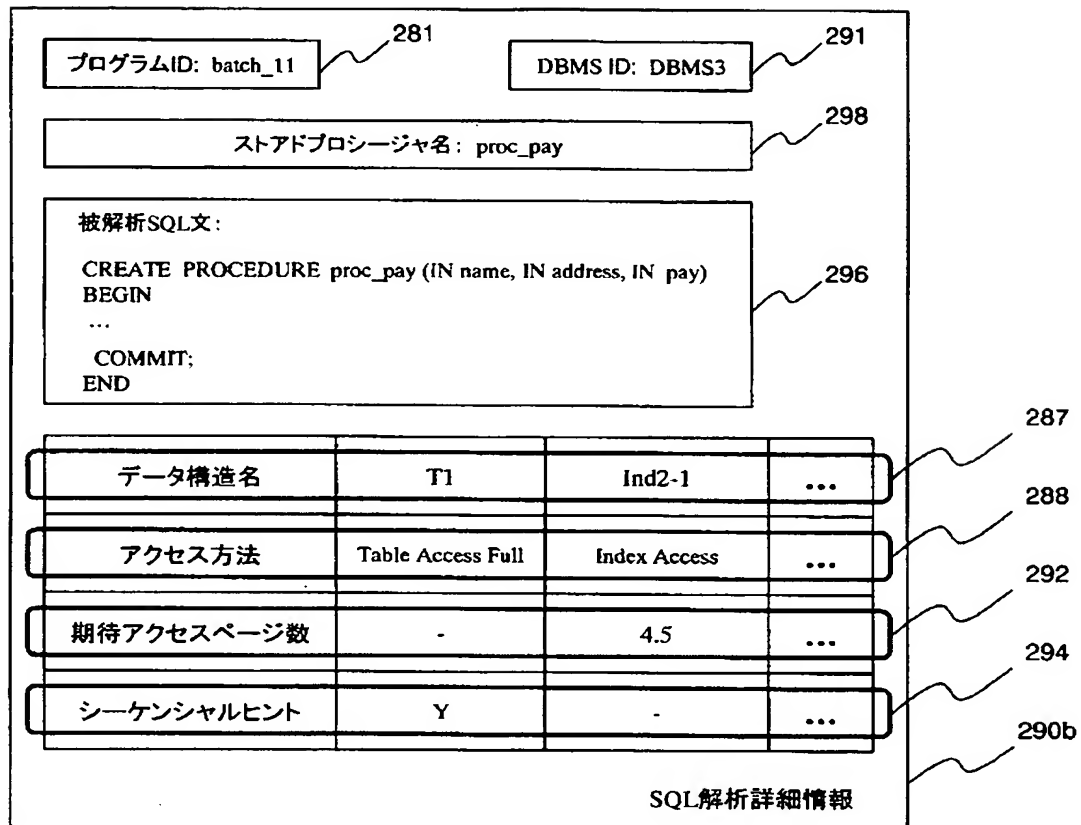
  ...

  COMMIT;
END
```

5200

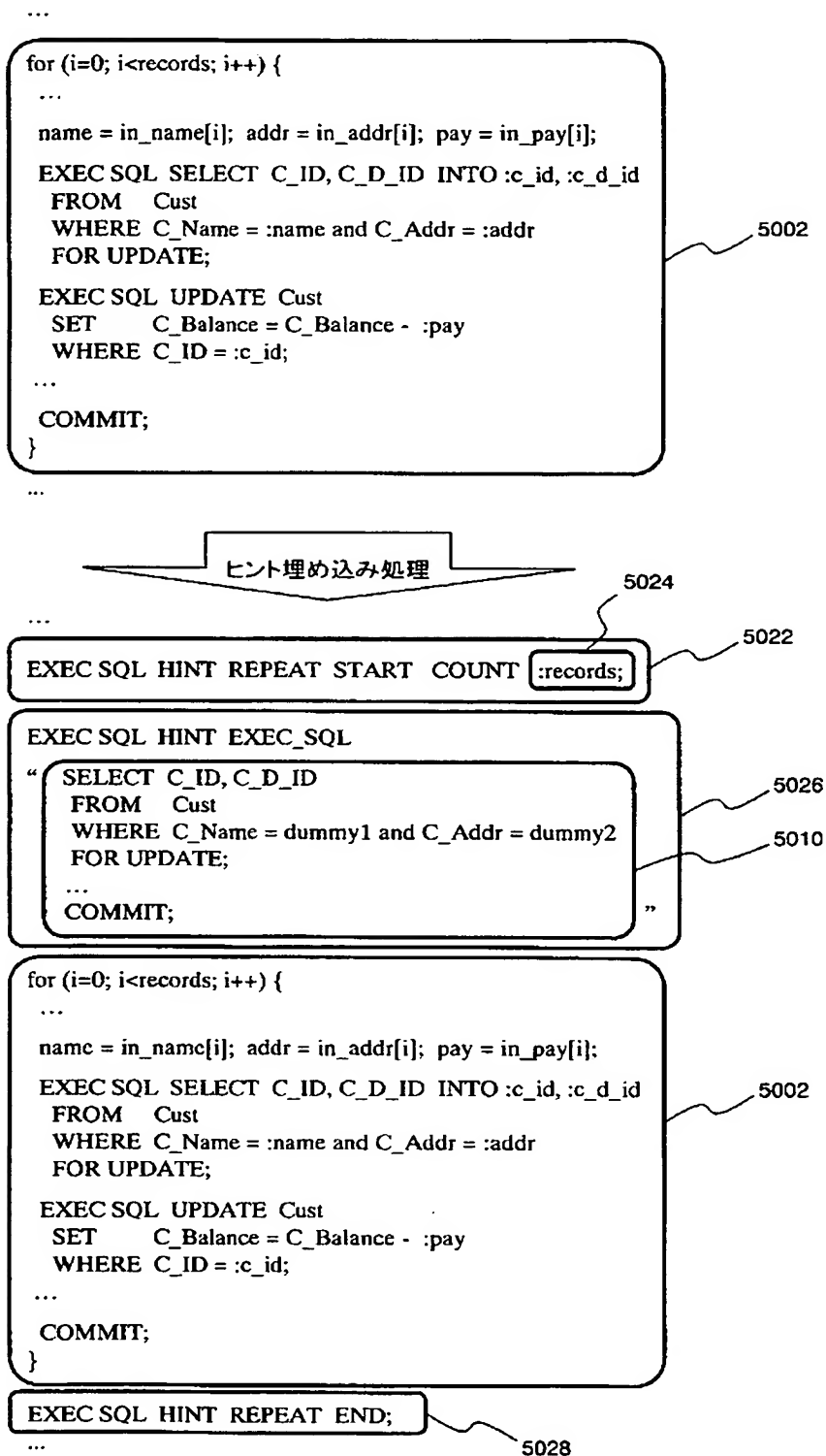
【図 24】

図24



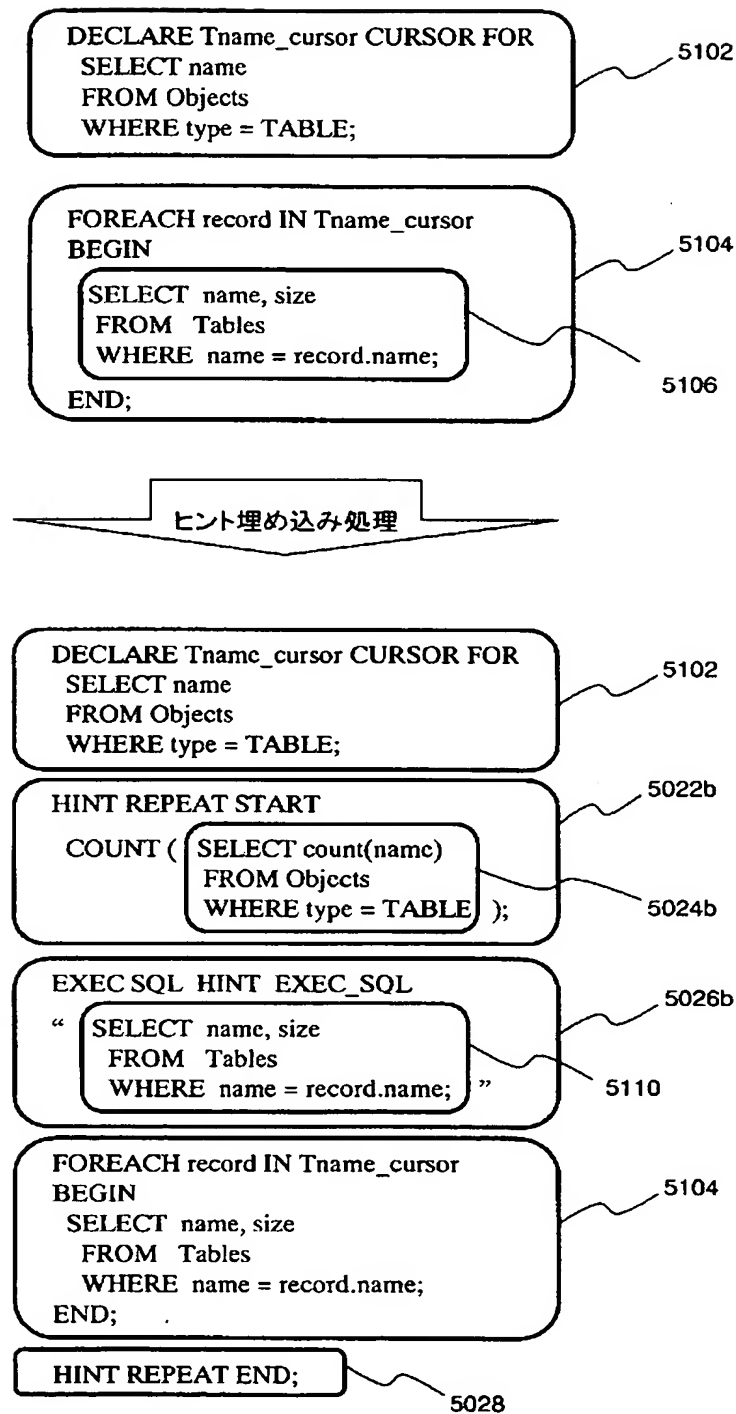
【図 25】

図25



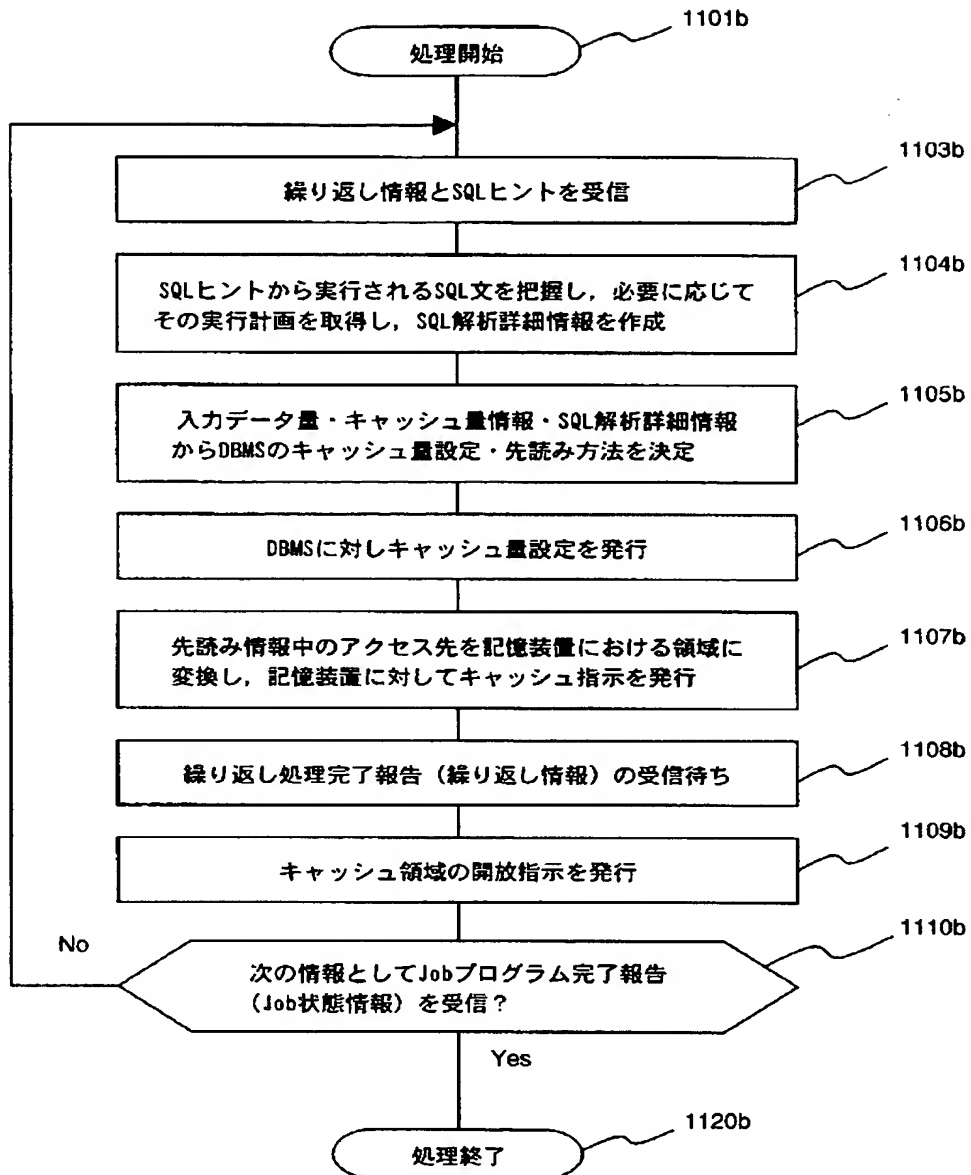
【図 26】

図26



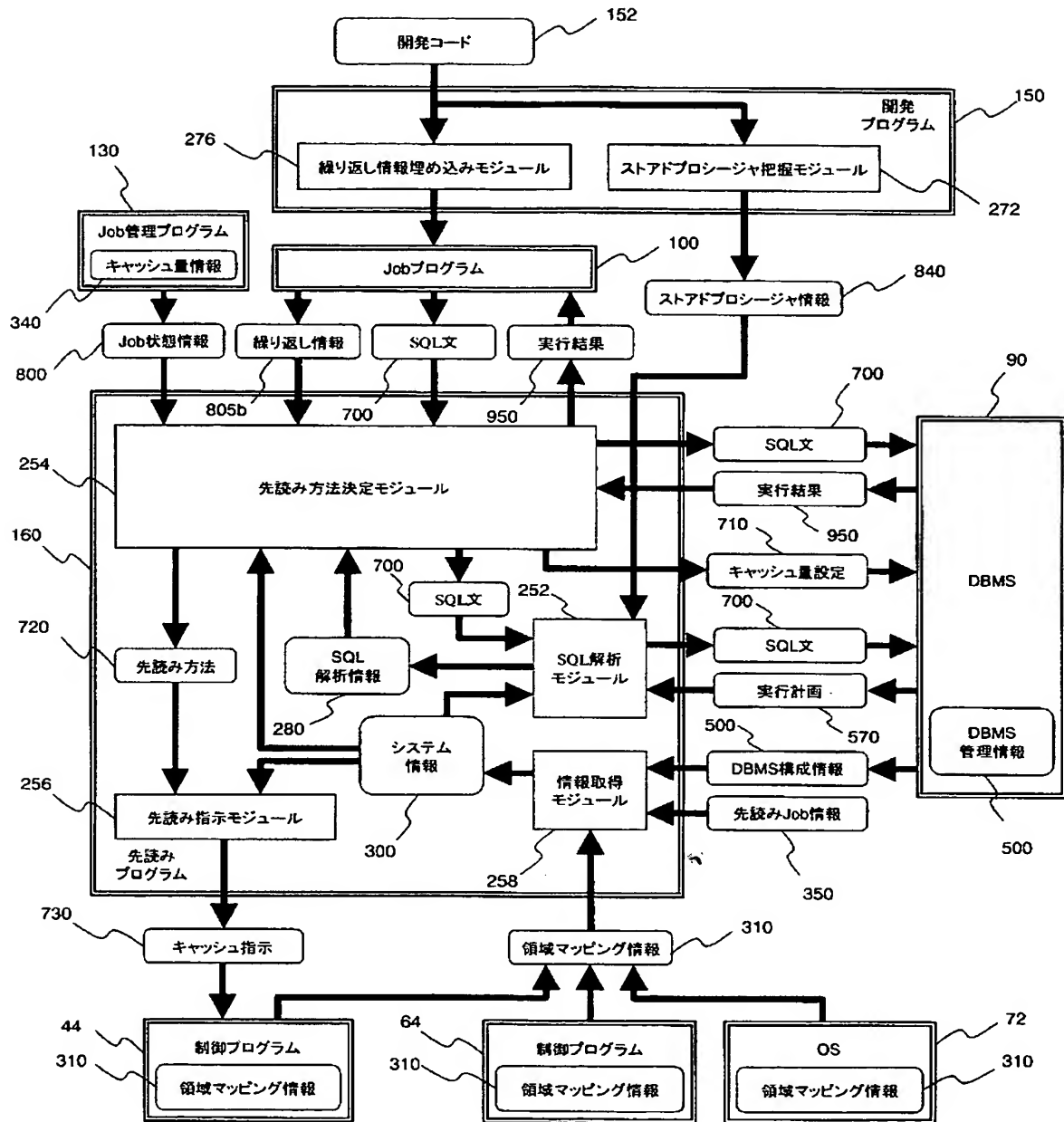
【図 27】

図27



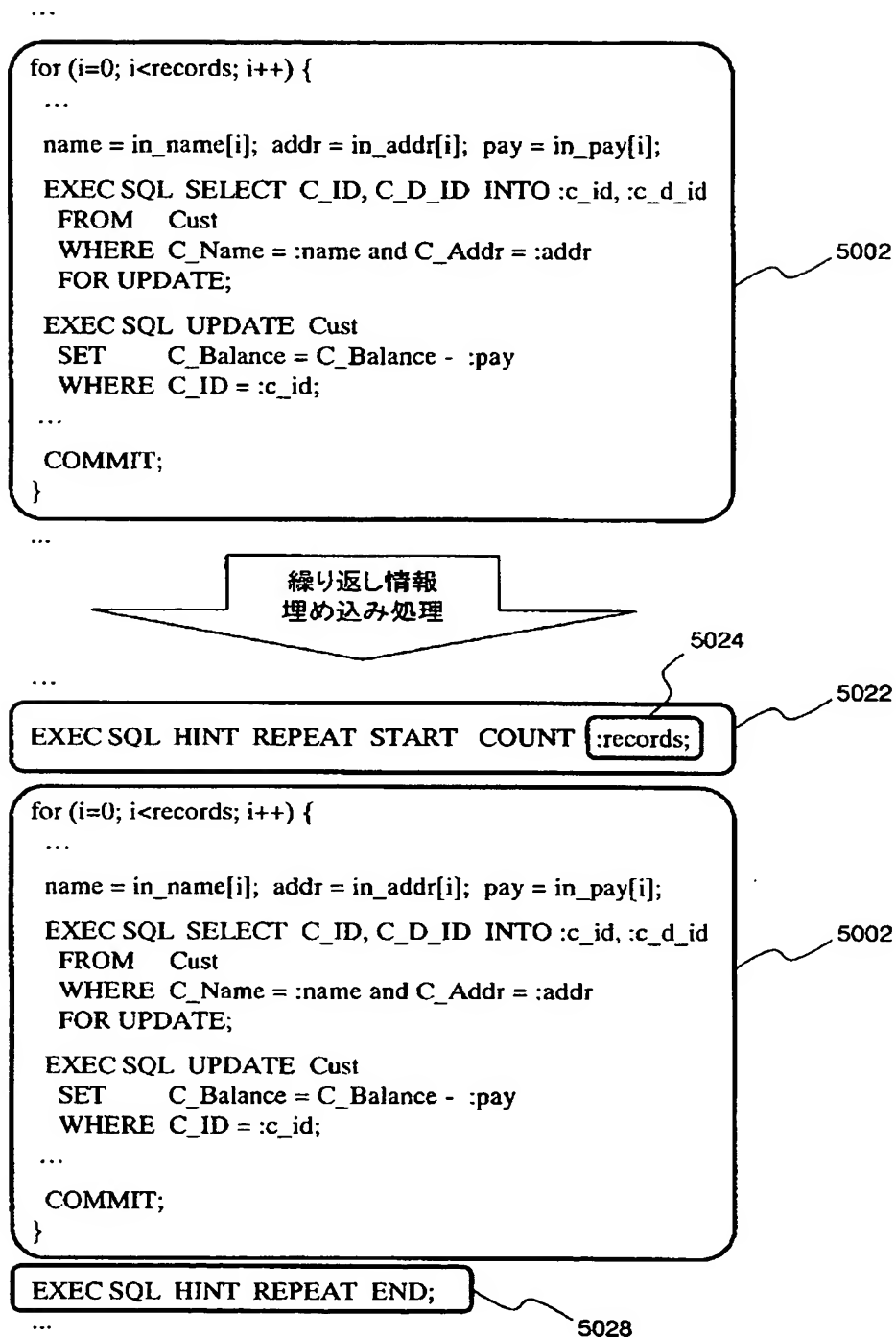
【図 28】

図28



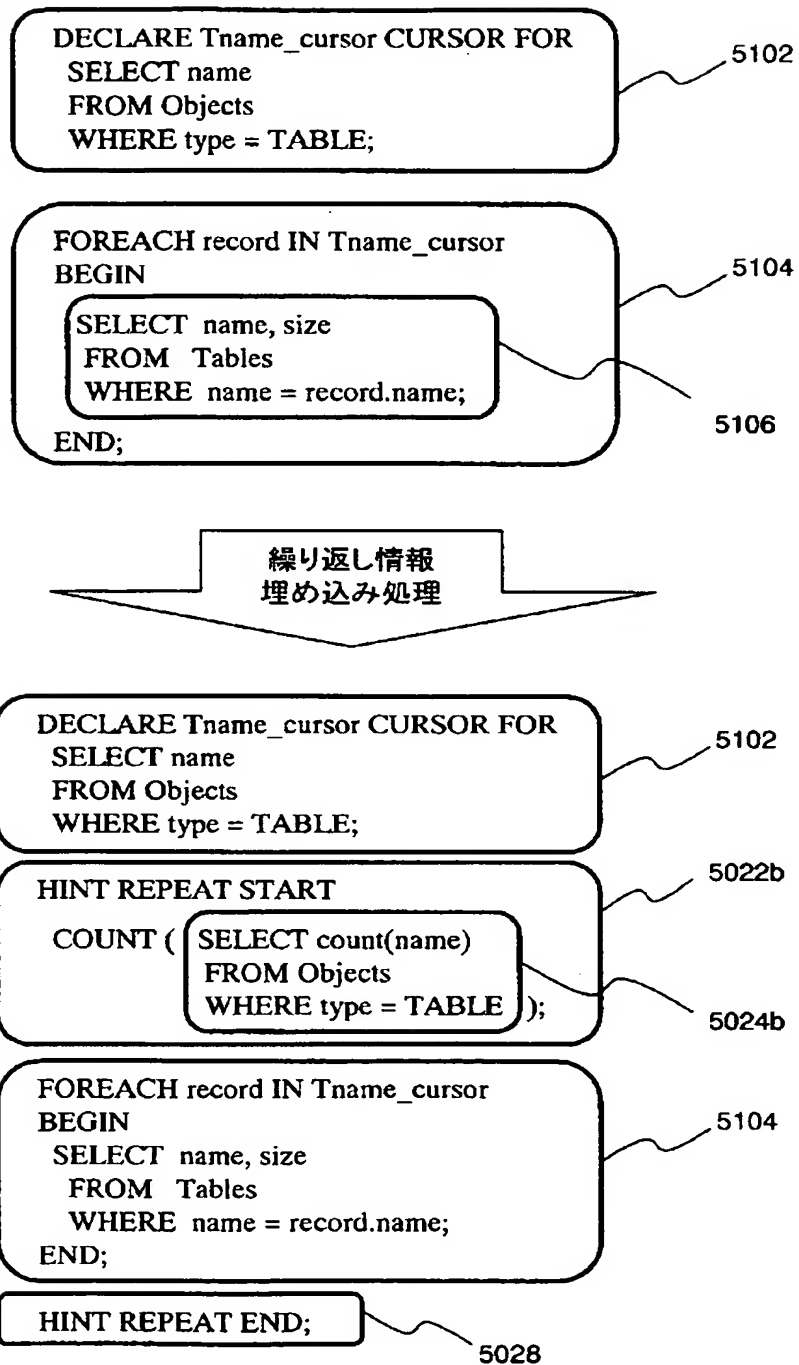
【図 29】

図29



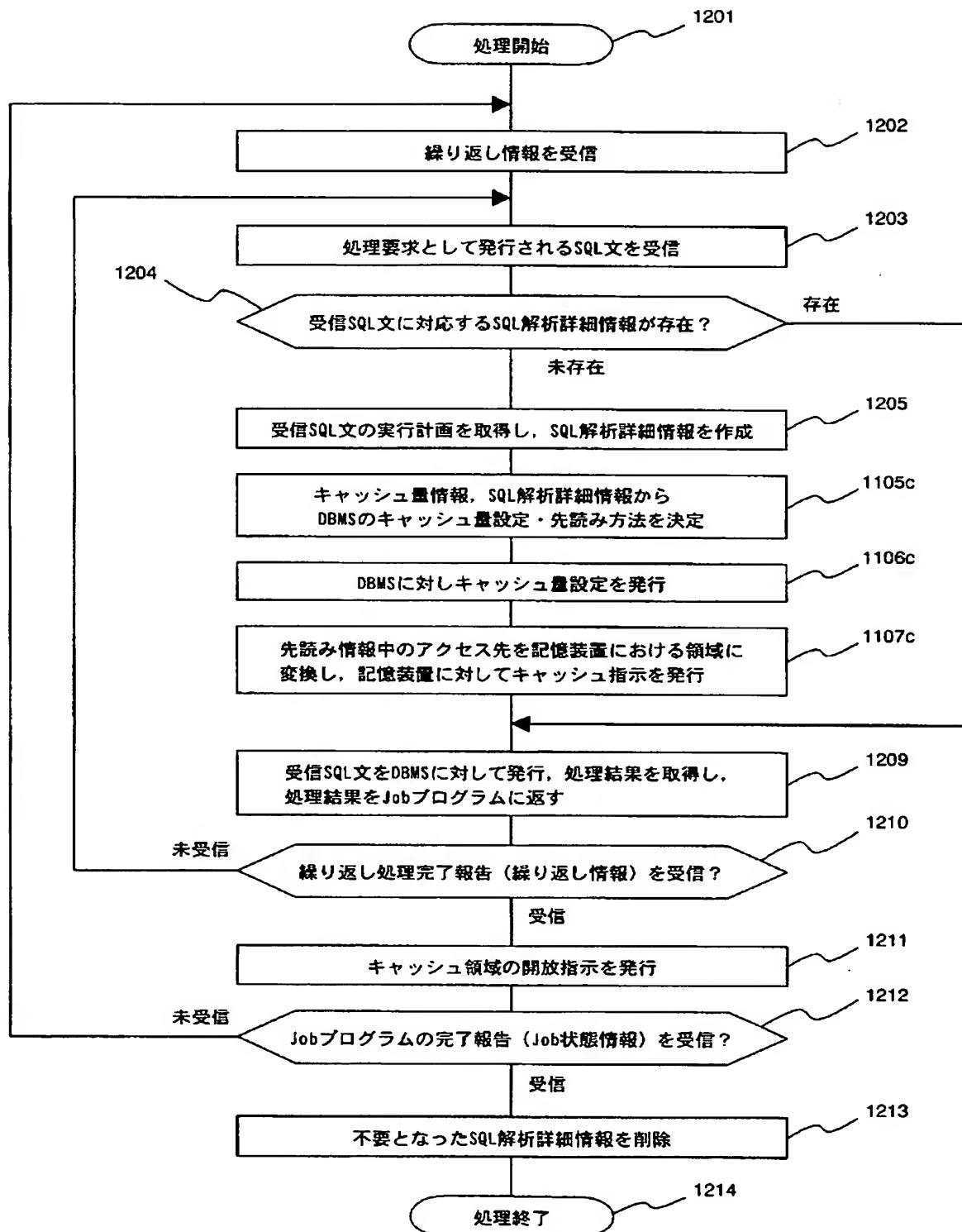
【図 30】

図30



【図 31】

図31



【書類名】 要約書

【要約】

【課題】 DBMSが稼動する計算機システムにおいて、同じ形のSQL文による処理が多数回、繰り返し実行される場合にデータの先読みを実施し、データのアクセス性能を向上させる。

【解決手段】 先読みプログラムが、繰り返し実行されるSQL文と、その処理の実行開始情報を取得し、それを基にデータの先読み指示を記憶装置に対して発行する。第一の方法では、繰り返し実行SQL文の取得と、処理内容の解析を先読みプログラムが事前に実施し、処理開始の通知を受け、事前解析を基に先読み指示を発行する。第二の方法では、繰り返し実行SQL文を処理開始時に先読みプログラムに与え、その解析結果を基に先読み指示を発行する。第三の方法では、先読みプログラムはDBMSのフロントエンドプログラムの的に振舞う。与えられたSQL文が繰り返し実行されるとして解析後、先読み指示を発行し、その後にDBMSに対してSQL文を転送する。

【選択図】 図 8

認定・付加情報

特許出願の番号	特願 2 0 0 3 - 0 8 6 8 2 9
受付番号	5 0 3 0 0 4 9 9 4 8 4
書類名	特許願
担当官	第七担当上席 0 0 9 6
作成日	平成 1 5 年 3 月 2 8 日

< 認定情報・付加情報 >

【提出日】	平成15年 3月27日
-------	-------------

次頁無

特願 2003-086829

出願人履歴情報

識別番号

[000005108]

1. 変更年月日

1990年 8月31日

[変更理由]

新規登録

住 所

東京都千代田区神田駿河台4丁目6番地

氏 名

株式会社日立製作所